UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA, MANAGUA

UNAN - MANAGUA

FACULTAD REGIONAL MULTIDISCIPLINARIA DE CHONTALES

FAREM-CHONTALES

PROGRAMA DE DOCTORADO EN MATEMÁTICA APLICADA

Tesis Para Optar Al Grado De Doctor En Matemática Aplicada

TRAVELING SALESMAN PROBLEM (TSP)
Diseño de Algoritmos Heurísticos y Metaheurísticos eficientes para resolver el Problema del Agente Viajero

Autor: M.Sc. José Jesús Mendoza Casanova

Tutor: M.Sc. - Ph.D. Antonio Parajón Guevara

Agosto, 2017

ANTONIO PARAJÓN GUEVARA, profesor titular del Departamento de Matemática de la Facultad de Educación e Idiomas de la Universidad Nacional Autónoma de Nicaragua, UNAN-Managua.

CERTIFICA que la presente memoria de investigación:

DISEÑO DE ALGORITMOS HEURÍSTICOS Y METAHEURÍSTICOS EFICIENTES PARA EL PROBLEMA DEL AGENTE VIAJERO

Ha sido realizada bajo su dirección en el PROGRAMA DE DOCTORADO EN MATEMÁTICA APLICADA por el Máster José Jesús Mendoza Casanova, y constituye su tesis para optar al grado de Doctor en Matemática Aplicada.

Y para que así conste, en cumplimiento con la normativa vigente de posgrado, autoriza su presentación ante la Facultad Regional Multidisciplinaria de Chontales (FAREM-Chontales) para que pueda ser tramitada su lectura y defensa pública.

Managua, Nicaragua, 21 de Agosto de 2017.

El DIRECTOR DE LA TESIS

Antonio Parajón Guevara, MS.c - Ph.D

Con mucho cariño,
a mi familia,
por su amor y soporte.

Agradecimientos

Quiero expresar mi agradecimiento:

A Dios.

Al Doctor Antonio Parajón Guevara, mi profesor de Programación Matemática y Optimización Combinatoria, tutor de este trabajo, sin el cual no hubiera podido llevar a cabo esta investigación.

Al Decano de la Facultad de Ciencias e Ingeniería, el maestro Marlon Leonel Díaz, por su gentileza y apoyo durante el programa del doctorado.

Al Maestro Álvaro Enrique Pavón Bonilla por su apoyo con el editor de textos científicos, LaTeX.

Al colectivo de Doctores de la Universidad Central **Marta Abreu** de Las Villas (UCLV), en Santa Clara, Cuba, por su apoyo incondicional.

A la Universidad Nacional Autónoma de Nicaragua, UNAN-Managua, y a la Dirección de Posgrado de la FAREM-Chontales, por haberme concedido la oportunidad de formar parte del Programa de Doctorado en Matemática Aplicada, por depositar en mí la confianza para la realización de este trabajo de investigación, y por los esfuerzos que realiza cada día por la formación continua de sus docentes.

A mis colegas doctorantes por sus valiosos aportes, sugerencias, fuerza y perseverancia que animan a seguir adelante; y al Decano Emilio López, por su juiciosa manera de motivarnos.

A todos ellos, sinceramente...

. ***muchísimas gracias.***.

Trayectoria Académica del Docente Investigador

José Jesús Mendoza Casanova, obtuvo su Licenciatura en Ciencias de la Educación con Mención en Matemática por la Universidad Nacional Autónoma de Nicaragua (UNAN-Managua-2005). Es Máster en Didácticas Específicas: Especialidad en Didáctica de la Matemática (2013) por la Universidad Autónoma de Barcelona, UAB. Máster en Matemática Pura, UNAN-Managua (2011). Máster en Matemática Aplicada, FAREM-Chontales (2017). Es Profesor Titular con Maestría en la UNAN-Managua, desempeñándose desde el año 2000 en las áreas de Álgebra, Análisis Matemático e Investigación de Operaciones. Realiza estudios de doctorado en Matemática Aplicada coordinado por la Universidad Central **Marta Abreu** de Las Villas, de la hermana república de Cuba. Desarrolló Tesis Doctoral, en Investigación de Operaciones, bajo la tutoría del Dr. Antonio Parajón Guevara. Línea de investigación sobre el Probema del Agente Viajero (TSP), que es un NP-Hard en Optimización Combinatoria, en la que se diseñaron algoritmos heurísticos y metaheurísticos eficiente para resolver el (TSP).

Resumen

El Problema del Agente Viajero (Traveling Salesman Problem, TSP) es aún un problema abierto en el área de conocimiento de la Programación Matemática. Desde los años cincuenta ha despertado mucho interés dentro de la comunidad científica por su forma sencilla de enunciarse y por su extrema dificultad en resolverse, aún teniendo a mano el desarrollo tecnológico con computadoras cada día más veloces y un sinnúmero de ideas para tratarlo, desde los aportes teóricos, hasta métodos de aproximación como Simulated Annealing, Greedy, Inteligencia Artificial, Simulación heurística, entre otros.

En esta investigación, la metodología que se ha utilizado para abordar el Problema del Agente Viajero ha sido la siguiente: estructurar la teoría referida a convexidad y poliedros sobre el TSP; diseñar algoritmos heurísticos tales como el Nearest Neighbor (Vecino Más Cercano) y el Greedy Randomized Adaptive Search Procedures-GRASP (Procedimientos de Búsqueda Basados en Funciones Voraces o Codiciosas), y metaheurísticos como el Tabu Search (Búsqueda Tabú) o el Scatter Search (Búsqueda Dispersa). Ambos tipos de algoritmos se implementaron en lenguaje C++ del ambiente de programación Microsoft Visual Studio 2008, y los resultados se compararon con los obtenidos mediante el uso de los softwares académicos Solver de Excel y WinQSB.

Finalmente, se compararon los resultados obtenidos por los algoritmos de esta investigación, con los encontrados por el Solver, WinQSB y los que aparecen en la TSPLIB, y se concluyó que de los dos algoritmos heurísticos y los dos metaheurísticos diseñados, resultaron muy satisfactorios los dos últimos en el balance entre eficacia y eficiencia. El mejor algoritmo para instancias pequeñas y medianas fue el Tabu Search, y para instancias grandes el Scatter Search.

Palabras claves: Investigación de Operaciones, Problema del Agente Viajero, Métodos Exactos, Métodos Heurísticos, Métodos Metaheurísticos.

Abstract

The Traveling Salesman Problem (TSP) is still an open problem in the area of Mathematica Programming knowledge. Since the 1950s it has aroused much interest with in the scientific community for its simple way of being uttered and for its extreme difficulty in solving, even having at hand the technological development with computers every day faster and a number of ideas to treat it, from the Theoretical contributions, to approximation methods such as Simulated Annealing, Greedy, Artificial Intelligence, Heuristic simulation, among others.

In this research, the methodology that has been used to address the Traveler Agent Problem has been the following: to structure the theory related to convexity and polyhedra over TSP; To design heuristic algorithms such as the Nearest Neighbor and Greedy Randomized Adaptive Search Procedures (GRASP), and metaheuristic such as Tabu Search or Scatter Search. Both types of algorithms were implemented in the C++ language of the Microsoft Visual Studio 2008 programming environment, and the results were compared with those obtained using the Excel and WinQSB academic softwares.

Finally, we compared the results obtained by the algorithms of this research, with those found by Solver, WinQSB and those that appear in the TSPLIB, and it was concluded that of the two heuristic algorithms and the two metaheuristics designed, the two were very satisfactory The balance between effectiveness and efficiency. The best algorithm for small and medium instances was the Tabu Search, and for large instances the Scatter Search.

Key words: Investigation Operations, Traveling Salesman Problem, Exact Methods, Heuristics Methods, Metaheuristics Methods.

Índice

1 ÁREA DE CONOCIMIENTO

La importancia de las investigaciones en matemáticas aplicadas radica en el influjo de las necesidades de otras disciplinas del saber humano, ya sea en la aplicación directa de resultados suficientemente desarrollados (teoremas, algoritmos, etc.) o bien en la construcción de modelos matemáticos (al menos parcialmente) para dar respuesta a los problemas planteados.

Las matemáticas han tenido y tienen una importancia creciente en la sociedad de cada tiempo, debido fundamentalmente a sus consecuencias. Aunque las aplicaciones más antiguas hay que buscarlas en la Física y en la Tecnología, recientemente es en la informática y las telecomunicaciones que se pueden encontrar dichas aplicaciones.

Para una apropiada descripción del área de conocimiento y línea de investigación en que se sitúa este trabajo, es necesario secuenciar su ubicación dentro de la matemática aplicada. La jerarquía de estas ramas son: programación matemática, programación lineal y programación lineal entera. Finalmente, en esta última se ubica el Problema de Agente Viajero (TSP), que además corresponde a la línea de investigación de optimización combinatoria, y que aún no ha sido resuelto por su complejidad computacional, es decir que corresponde a la clase de problemas NP-Hard.

Por tal razón, a lo largo de la historia los investigadores de optimización se han dado a la tarea de descubrir métodos más eficientes para encontrar mejores soluciones. De modo que, en esta tesis se proponen algoritmos heurísticos y metaheurísticos diseñados para resolver el Problema del Agente Viajero.

2 TEMA DELIMITADO

DISEÑO DE ALGORITMOS HEURÍSTICOS Y METAHEURÍSTICOS

EFICIENTES PARA RESOLVER EL PROBLEMA DEL AGENTE VIAJERO

3 INTRODUCCIÓN

La investigación es una actividad orientada a la obtención de nuevos conocimientos y su aplicación para la solución de problemas sociales, o bien cuestiones de carácter científico. En ello radica su importancia. En el caso de la Matemática, la investigación se produce en este momento en todos los países del mundo, con las limitaciones propias del potencial de cada país, pero sin los inconvenientes de las barreras ideológicas o culturales. El trabajo del matemático en este campo se da tanto a nivel teórico como a nivel aplicado. Y la importancia de este tipo de trabajos radica en el influjo de las necesidades de otras disciplinas del saber humano y los problemas surgidos de su propio contexto.

No bastaría el tamaño de este documento para mencionar los grandes aportes de la matemática a nivel aplicado. Sin embargo, se deben citar algunos ejemplos de impacto de esta ciencia en la sociedad moderna: la teoría de la computación está basada en la Matemática Discreta y de esa relación nació la Teoría de la Complejidad; modelos integro-diferenciales han sido extendidos hacia la Epidemiología y Genética Molecular, ello ha permitido el diseño de medicamentos y su administración. Otro ejemplo de su impacto en la sociedad moderna, popularizado por producciones literarias y cinematográficas, son los aportes que hiciera a las teorías económicas el matemático estadounidense, y Premio Nobel de Economía 1994, John Nash; su vida es retratada en la película “A Beautiful Mind", basada en la biografía escrita por Sylvia Nasar.

En cuanto a los grandes aportes de la matemática en el ámbito teórico, el mejor ejemplo es su relación simbiótica con la Física, ello merece un capítulo aparte: son muchos los volúmenes que podrían escribirse para exponer esa relación. Un ápice de ello lo encontramos en los siguientes hechos. La Física moderna se vería en serios aprietos sin los aportes de la Matemática; por ejemplo, la Teoría de la Relatividad no tendría el desarrollo y profundo impacto científico y tecnológico sin las bases proporcionadas por las geometrías no euclidianas.

En Nicaragua, también se hacen esfuerzos con el desarrollo de investigaciones matemáticas. Las universidades son pioneras en estos tipos de trabajos científicos. Las instituciones de educación superior más importantes y más grandes del país, tienen incorporadas academias de jóvenes talentos, la carrera de matemática, maestrías y hasta doctorados en esta disciplina. Actualmente, en la UNAN-Managua se está desarrollando el Programa de Doctorado de Matemática Aplicada, bajo la coordinación de la Universidad Central “Marta Abreu” de Las Villas, Cuba. Los grandes esfuerzos que nuestro país realiza tienen como finalidad el beneficio de la sociedad.

Esta investigación es de carácter aplicado, y se realiza un estudio sobre el Problema del Agente Viajero (TSP), uno de los problemas más estudiados en Programación Matemática. La importancia de investigar sobre este problema se debe a que es un problema abierto, difícil de resolver, pero que tiene muchas aplicaciones prácticas en la industria, mercados bursátiles, turismo, planificación, logística, fabricación de circuitos electrónicos, secuencia de ADN, etc.

Este problema de apariencia sencilla es famoso por su gran complejidad computacional. Como se verá más adelante, se encuadra dentro de la categoría NP-duro por lo que, al día de hoy, no se ha encontrado ningún algoritmo que logre resolverlo en un tiempo polinómico. Se formula de la siguiente manera: un comerciante debe visitar varios clientes y desea conocer cuál es el camino de mínima distancia que, partiendo de su lugar de trabajo, vaya a todas las ciudades y regrese a la ciudad de origen.

Por tanto, en esta investigación se van a diseñar algoritmos heurísticos y metaheurísticos para resolver el Problema del Agente Viajero. Para ello se ha estructurado en quince apartados, en los tres primeros se establece el área de conocimiento y línea de investigación, el tema delimitado y la introducción que describe el cuerpo de esta memoria.

En el cuarto apartado de este trabajo se expone el estado del arte del problema que aquí se está abordando. Primero se hace una breve reseña histórica para comprender el génesis de la investigación de operaciones, sus aplicaciones y la metodología científica aplicada en el estudio. Finalmente, se explica el Problema del Agente Viajero que es nuestro principal interés. Se citan las investigaciones sobresalientes por haber marcado hito en cuanto a su sencillez, originalidad y garantía de desempeño.

En el apartado cinco se plantea en forma natural el problema que ha motivado esta investigación, se explica por qué este corresponde a la categoría de problemas NP-duros o difíciles de resolver y finalmente se hace la formulación detallada del TSP. Asimismo, en el siguiente apartado se precisa el potencial de este estudio mediante la justificación y viabilidad del mismo.

El objetivo general y los objetivos específicos correspondientes se presentan en el apartado siete.

Los antecedentes del TSP se señalan en el apartado ocho de esta memoria. Aquí se describe y se analiza un espacio literario relevante sobre el Problema del Vendedor Viajero en términos de contenido, clases de TSP, métodos y campos de inspiración. Los datos empleados provinieron de los trabajos más citados en Scopus (base de datos bibliográfica de resúmenes y citas de artículos de revistas científicas) sobre el TSP, tanto a través de la historia como en el período 2006-2010.

El apartado nueve corresponde al marco teórico, primero se presentan algunos resultandos importantes de la Combinatoria Poliédrica. Luego se aborda la teoría sobre el Problema del Agente Viajero. Se describe la complejidad computacional del TSP, que lo convierte en un problema NP-duro o difícil de resolver. Finalmente, se explican los métodos de solución del TSP.

En el apartado diez se refleja la metodología que se siguió en esta investigación. Se describe la revisión del estado del arte, el diseño de los algoritmos heurísticos y metaheurísticos, implementación y la comparación de los resultados, usando las instancias de la TSPLIB, esta biblioteca virtual permitirá determinar la calidad de los algoritmos. Finalmente, se presenta el cronograma de trabajo que se ha seguido en esta investigación.

Los resultados y el análisis de los mismos se exponen en el apartado once. Primero se muestra la eficiencia de los algoritmos con las instancias simétricas, comparando los resultados con las salidas que nos proporcionan otros softwares de optimización como el Solver de Excel y el muy conocido WinQSB. Luego se presentan las tablas de análisis de un Scatter Search (Búsqueda Dispersa) diseñado para resolver el Problema del Agente Viajero.

Por último, las conclusiones, perspectivas de futuro, bibliografía y anexos aparecen en los apartados doce, trece, catorce y quince, respectivamente.

4 ESTADO DEL ARTE

4.1 Reseña histórica de la Investigación de Operaciones (IO)

A lo largo de la historia es frecuente encontrar una estrecha colaboración entre científicos y militares con el fin de dictaminar la decisión óptima en la batalla e intentar obtener la victoria. Es por esto que muchos expertos en la materia consideran el inicio de la Investigación Operativa en el siglo III A.C., durante la II Guerra Púnica, con el análisis y solución que Arquímedes propuso para la defensa de la ciudad de Siracusa, sitiada por los romanos. Entre sus inventos se encontraban la catapulta, y un sistema de espejos con el que incendiaba las embarcaciones enemigas al enfocarlas con los rayos del sol.

En 1503, Leonardo da Vinci participó como ingeniero en la guerra contra Pisa ya que conocía técnicas para realizar bombardeos, construir barcos, vehículos acorazados, cañones, catapultas, y otras máquinas bélicas.

El concepto de Investigación de Operaciones nació durante la Primera Guerra Mundial en Inglaterra entre los años 1914-1915, cuando F. W. Lanchester intentó tratar cuantitativamente las operaciones militares, obteniendo ecuaciones que relacionaban el resultado de una batalla en función de la fuerza numérica relativa de los combatientes y de su capacidad relativa de fuego. Lanchester modeló una situación que involucraba opciones estratégicas, y después probó ese modelo contra la situación real. Éste procedimiento es el que los investigadores de operaciones han venido practicando desde entonces.

Entre los años 1914-1015, Tomás Alva Edison en los Estados Unidos de América, estudió el proceso de la guerra antisubmarina. Efectuó un análisis estadístico para desarrollar maniobras mediante las cuales los barcos pudieran evadir y destruir a los submarinos.

En 1917, el matemático Danés A. K. Erlang, que trabajaba en la compañía telefónica de Copenhage, publicó el trabajo "Soluciones a Algunos Problemas en la Teoría de Probabilidades" importantes en las centrales telefónicas automáticas, contenía fórmulas de tiempo de espera que más tardes fueron empleadas por la Oficina Postal Británica para calcular el número de circuitos necesarios.

En 1915, Ford W. Harris describió el primer modelo sobre el tamaño de lote económico de inventario, posteriormente contribuyeron al desarrollo de modelos de control de inventarios H. S. Owen en 1925, Benjamín Cooper en 1926, R.H. Wilson en 1926 y W. A. Mueller en 1927. Las técnicas matemáticas del control de inventarios son de las más antiguas herramientas de la Investigación de Operaciones.

En 1939, el matemático ruso Leonid Vitálievich Kantoróvich y el holandés Tjalling Charles Koopmans, desarrollaron la teoría matemática llamada "Programación Lineal", por la que les fue concedido el Premio Nobel de Economía. Dos años después, estudiaron de forma independiente el problema del transporte por primera vez, conociéndose este tipo de problemas como problema de Koopmans-Kantoróvich. Para su solución, emplearon métodos geométricos que están relacionados con la teoría de convexidad de Minkowski.

Como consecuencia del ingreso de Inglaterra a la Segunda Guerra Mundial dos años antes que Estados Unidos, en 1939 existía un núcleo de una organización Británica de Investigación de Operaciones y sus principales aportes fueron: El mejoramiento del sistema de radar, el cañoneo antiaéreo, en la guerra antisubmarina, en la defensa de la población civil, en el diseño del tamaño de los convoy y en la conducción de ataques de bombardeo sobre Alemania.

El grupo de Investigación de Operaciones con mayor publicidad fue el denominado "Circo de Blackett" dirigido por el profesor P.M.S. Blackett de la Universidad de Manchester, ministro de la Royal Society, laureado Nobel y ex-oficial naval. El grupo estaba conformado por tres fisiologistas, dos físicos matemáticos, un astrofísico, un oficial del ejército, un topógrafo, un físico general y dos matemáticos. El valor del enfoque del equipo heterogéneo fue de éxito notorio.

Estados Unidos (EEUU), al unirse a la Guerra en 1942, comenzó a aplicar técnicas de Investigación de Operaciones militarmente, y unos años más tarde, en 1947, formó un grupo de trabajo dedicado a mejorar los procesos de planificación a gran escala: el proyecto SCOOP (Scientific Computation Of Optimum Programs). En dicho grupo se encontraba trabajando George Bernard Dantzig, quien desarrolló en 1947 el algoritmo del Método Simplex.

Después de la Segunda Guerra Mundial, tanto el ejército como la fuerza aérea de los Estados Unidos de Norte América, continuaron con los grupos de Investigación de Operaciones pero las técnicas desarrolladas empezaron a ser usadas en la planeación de los negocios. La industria debía renovar su producción y organización para servir rápidamente a las necesidades en tiempos de paz. En 1950 se organizó la Operations Research Society of América (ORSA) y The Institute of Management Science (TIMS).

Desde 1952 ORSA publica la revista Operations Research y desde 1953 TIMS publica su revista Management Science. Desde la década de los 70 (s) las dos sociedades publican la revista trimestral Interfaces con trabajos y artículos relacionados con los problemas operacionales del uso de la ciencia administrativa y la investigación de Operaciones. En Inglaterra se formó en 1948 el Operational

Research Club quien cambió su nombre posteriormente a la Operational Research Society of the United Kingdom y para 1950 crearon la revista Operational Research Quarterly. Más recientemente se han formado sociedades de Investigación de Operaciones en Francia, Italia, Israel y Austria.

No obstante, el desarrollo de la Programación Lineal ocurrió hacia 1760 cuando los economistas empezaron a describir sistemas económicos en términos matemáticos. En ese mismo periodo el profesor de Harvard, y premio Nobel de economía (1973), Wassily Leontieff desarrolló un modelo de programación Lineal que representaba la totalidad de la economía de los Estados Unidos de Norte América.

Durante y después de la Segunda Guerra Mundial se formaron muchos grupos de investigación de operaciones. En la actualidad las principales revistas científicas que se ocupan de la investigación de operaciones, modelado y simulación son: European Journal of Operational Research (EJOR, fundada en 1977), Journal of the Operational Research Society (JORS, 1950), Informs Journal on Computing (JOC, 1987), Journal of Combinatorial Optimization (Springer, 1998), Operations research (Dialnet, 2003), etc.

4.2 La Investigación de Operaciones (IO)

El término IO se utiliza por primera vez en el año 1939 durante la Segunda Guerra Mundial, específicamente cuando surge la necesidad de investigar las operaciones tácticas y estratégicas de la defensa aérea, ante la incorporación de un nuevo radar, en oportunidad de los ataques alemanes a Gran Bretaña. El avance acelerado de la tecnología militar hace que los ejecutivos y administradores militares británicos deban recurrir a los científicos, en pos de apoyo y orientación en la planificación de su defensa. El éxito de un pequeño grupo de científicos que trabajaron en conjunto con el ejecutivo militar a cargo de las operaciones en la "línea", derivó en una mayor demanda de sus servicios y la extensión del uso de la metodología a USA, Canadá y Francia entre otros.

Sin embargo, el origen de la Investigación Operativa puede considerarse como anterior a la Revolución Industrial, aunque fue durante este período que comienzan a originarse los problemas tipo que la Investigación Operativa trata de resolver.

Los problemas tipos en la investigación de operaciones son: asignación de recursos escasos, ordenamiento, secuenciación y coordinación de tareas, líneas de espera, mantenimiento y reemplazo de equipos, inventarios, costos y tiempos, y gestión de proyectos.

La IO ofrece herramientas cuantitativas para la toma de decisiones que resuelven los problemas diarios de un negocio o sirven para tomar decisiones en la planeación a corto o largo plazo, sea el negocio de carácter gubernamental, producción, servicios, gremial o cooperativo.

En la investigación de operaciones se aplican los siguientes seis pasos metodológicos científicos, a saber: análisis y definición del problema, desarrollo del modelo, selección de datos de entrada, obtención de una solución, limitaciones del modelo y la solución, y utilización del modelo.

4.3 Investigaciones Relevantes Sobre el TSP

El Problema del Agente Viajero o TSP por sus siglas en inglés (Traveling Salesman Problem), es un problema clásico y abierto de la optimización combinatoria una de las principales subdisciplinas de la investigación de operaciones.

El TSP se puede formular de la siguiente manera: "Dado un conjunto de ciudades, de las cuales se conoce para cada par de ellas, la distancia que las separa, un agente viajero ha de partir de una ciudad de origen y debe visitar exactamente una vez cada ciudad del conjunto, y retornar al punto de partida". Un recorrido con estas características, es conocido dentro de este contexto, como un tour o ciclo hamiltoniano. El problema consiste en encontrar el tour para el cual la distancia total recorrida sea mínima.

Como se ha visto anteriormente, el problema en sí es fácil de formular, sin embargo pertenece a una clase de problemas muy difíciles de resolver, en el entendido que resolver significa hallar la solución óptima. Su dificultad radica en el hecho de que para cualquier algoritmo utilizado en la búsqueda de la solución óptima, éste emplea un tiempo de cómputo que crece exponencialmente a medida que aumenta la cantidad de nodos en el problema. Por esta razón surge la necesidad de aplicar métodos heurísticos para obtener, al menos, soluciones factibles de alta calidad, y cuando la heurística ha fracasado en dar resultados efectivos, entonces será necesaria la aplicación de métodos de aproximación metaheurísticos.

Entre las investigaciones para resolver el TSP sobresalen, mayormente, las que han marcado un hito por su sencillez, originalidad y garantía de desempeño:

- El método Simplex, desarrollado por Dantzig en 1947, busca el óptimo mediante la inspección de los vértices, lo que implica que el tiempo de cómputo crezca exponencialmente. Inicia en una solución de entrada y se mueve entre los nodos vecinos para obtener una mejor solución.

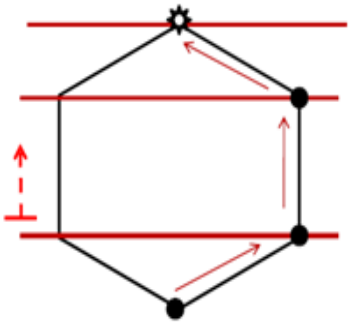


Figura 1: Método Simplex

- El método del Elipsoide, creado por Khachiyan en 1979, consiste en ir encerrando en cada paso la región factible en un elipsoide, la cual cada vez irá quedando más pequeña, tanto es así que converge. Aquí, el tiempo de cómputo es polinomial. Aunque este tiempo sea mejor que el exponencial, en la práctica resulta muy difícil trabajar con este método, pues involucra los conceptos de serie y convergencia, que resultan ser muy complicados en el ámbito computacional.

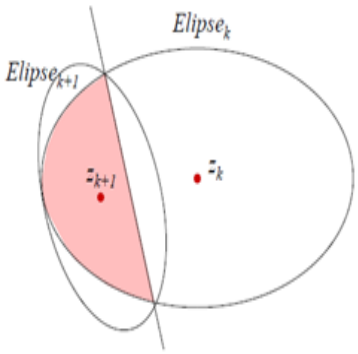


Figura 2: Método del Elipsoide

- El método del Punto Interior, establecido por Karmarkar en 1984, es más eficiente que el Simplex y busca desde el interior del dominio para llegar al punto óptimo. Ha sido bien estudiado, su tiempo de cómputo es polinomial. Sin embargo en la práctica esto no se ha plasmado del todo en los solvers debido a lo muy depurado, estudiado y optimizado que está el Simplex.

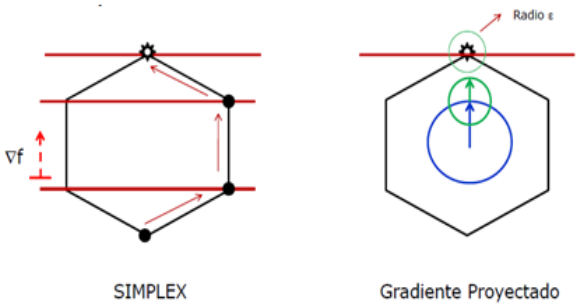


Figura 3: Método del Punto Interior

- El método de barreras logarítmicas permite avanzar, por el espacio de soluciones factibles hasta acercarse al óptimo. Esto se logra definiendo un factor μ que representa una barrera.

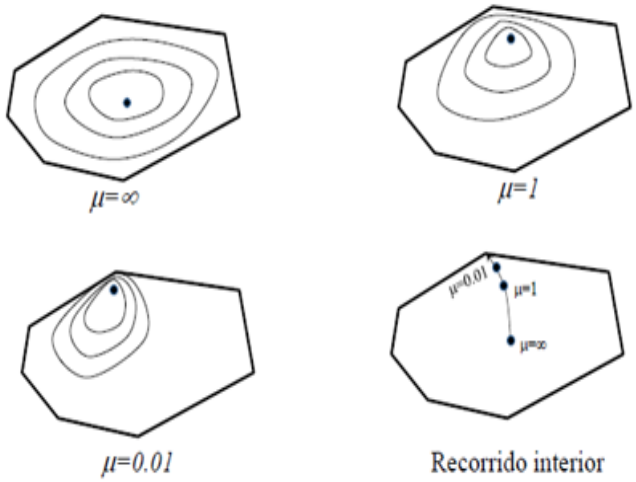


Figura 4: Método de Barreras Logarímicas

En virtud de lo anterior las expectativas para esta investigación radican en el diseño de un algoritmo, comparable con los resultados de la TSPLIB, a través de la combinación de métodos heurísticos y metaheurísticos propuestos por otros investigadores, y de autoría propia.

5 PLANTEAMIENTO DEL PROBLEMA

5.1 Formulación del Problema del Agente Viajero (TSP)

El Problema del Agente Viajero se formula de la siguiente manera:

Dado un conjunto de n ciudades, de las cuales se conoce para cada par de ellas, la distancia que las separa, un agente viajero ha de partir de una ciudad de origen y debe visitar exactamente una vez cada ciudad del conjunto, y retornar al punto de partida. Un recorrido con estas características es llamado, dentro de este contexto, ciclo hamiltoniano o tour. El problema consiste en encontrar el ciclo para el cual la distancia total recorrida sea mínima.

No obstante, para enunciar el problema formalmente se introducen las siguientes terminologías: Sea un grafo $G = (V, A, C)$ donde V es el conjunto de vértices, A es el de aristas y C_{ij} es la matriz de costes. Esto es, C_{ij} es el coste o distancia de la arista (i, j).

- Un camino (o cadena) es una sucesión de aristas $(e_1, e_2, ..., e_k)$ en donde el vértice final de cada arista coincide con el inicial de la siguiente. También puede representarse por la sucesión de vértices utilizados.
- Un **camino** es simple o elemental si no utiliza el mismo vértice más de una vez.
- Un **ciclo** es un camino $(e_1, e_2, ..., e_k)$ en el que el vértice final de e_k coincide con el inicial de e_1.
- Un **ciclo** es simple si lo es el camino que lo define.
- Un **subtour** es un ciclo simple que no pasa por todos los vértices del grafo.
- Un **tour** o ciclo hamiltoniano es un ciclo simple que pasa por todos los vértices del grafo.

El Problema del Agente Viajero consiste en determinar un tour de coste mínimo. La Siguiente figura muestra un grafo de 8 vértices en el que aparece destacado un ciclo hamiltoniano.

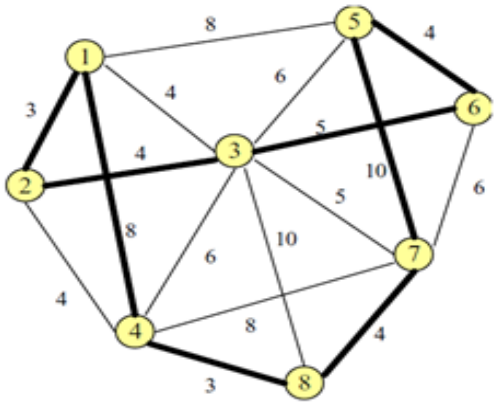


Figura 5: Ciclo Hamiltoniano

Es decir que el TSP consiste en encontrar el camino mínimo que recorre todos los nodos de un grafo, saliendo de uno de ellos y retornando al mismo vértice de partida. El problema se puede representar como un grafo completo orientado o bien como una permutación. Los nodos representan las ciudades y los valores asociados a las aristas, son las distancias entre ellas, y por tanto, nunca pueden ser negativos.

5.2 ¿Por qué el TSP es un NP-Hard?

Habitualmente, cuando se habla del Problema del Agente viajero se hace referencia a la dificultad de su resolución, pero lo cierto es que, a pesar de no haber encontrado todavía un algoritmo “eficiente”, no se puede afirmar que éste no exista.

En ese sentido, se puede asegurar que el TSP tiene solución, pues siempre pueden, en teoría, evaluarse todas las posibles soluciones y escoger la de menor coste (Método de Fuerza Bruta). El problema es que el número de posibles soluciones aumenta de manera significativa cuando aumenta el tamaño del problema. El número de ciclos posibles puede calcularse de manera sencilla: el origen viene determinado por lo que restarán $(n-1)$ puntos para empezar, a continuación deberemos elegir cualquiera de los $(n-2)$ restantes y así sucesivamente. De esta forma, multiplicando todas estas cantidades se obtiene el número total de rutas o soluciones posibles:

$$(n-1)! = (n-1)(n-2)(n-3) \cdot 3 \cdot 2 \cdot 1$$

Por lo tanto, el método directo implica evaluar $(n-1)!$ soluciones posibles. En el caso particular de 10 ciudades esto significaría 362880 evaluaciones, es decir, para un problema no excesivamente grande el número de pruebas aumenta considerablemente. Así, aunque el problema puede resolverse

en un número finito de pasos, esto no es suficiente y deberán buscarse otros algoritmos inteligentes que reduzcan este número de pruebas.

De acuerdo con lo anterior, para evaluar y comparar algoritmos podría utilizarse el número de iteraciones (o pruebas) que requieren para alcanzar una solución. En el caso del método directo, el número de pasos es siempre $(n-1)!$, pero existen otros algoritmos para los que, debido a su naturaleza, es muy difícil, o incluso imposible, calcular este número de pasos. Se debe, entonces, buscar otro método de comparación. El método que generalmente se emplea consiste en comparar el tiempo que los métodos emplean en encontrar dicha solución. El tiempo que se asocia a cada algoritmo debe ser una función de (el tamaño del problema), que represente una cota del tiempo máximo que necesite para alcanzar la solución. Como ocurre con todo método, éste también tiene sus desventajas: se está juzgando un algoritmo según el tiempo que necesita para resolver el problema que peor resuelve, pero puede ocurrir que generalmente necesite un tiempo muy inferior a esta cota.

En el año 1962, M. Held y R. Karp descubrieron un nuevo algoritmo que, basado en programación dinámica, requiere de un tiempo proporcional a $n^2 2^n$ (es destacable, además, que este algoritmo tiene el mejor tiempo computacional de entre todos los algoritmos capaces de resolver cualquier TSP descritos hasta el momento). Utilizando la técnica de comparación definida anteriormente, podemos concluir que este nuevo método es significativamente mejor que el método directo (para el caso particular de $n = 10$ el número de soluciones se reduce de $(10-1)! = 9! = 362880$ a 102400). Mientras que ningún ordenador del mundo era capaz de resolver un problema del viajante de 50 ciudades con el método directo, con el algoritmo definido por Karp esto se hizo posible. No obstante, si el número de ciudades se duplica, la tecnología actual tampoco permite calcular el óptimo a través de ese algoritmo. Entonces, ¿qué es lo que hace a un algoritmo "bueno"?

El matemático J. Edmonds propuso una definición formal de lo que es un "buen" algoritmo de resolución. Para él, un algoritmo puede clasificarse como bueno o eficiente si el tiempo necesario para que alcance el máximo, entendido según la definición anterior, es $K \cdot n^c$, siendo K y c dos números constantes. Generalmente, la constante K se elimina y se habla de algoritmos $O(n^c)$. Según las definiciones anteriores, el método propuesto por Held y Karp es un algoritmo $O(n^2 2^n)$ y, por tanto, no es un buen algoritmo. Dado que el término "bueno" tiene muchos significados, muchos investigadores prefieren denominar estos algoritmos como algoritmos de tiempo polinómico. De esta forma, los problemas pueden clasificarse como fáciles o difíciles según exista o no un método eficiente para resolverlos.

Una cuestión muy importante, que fue planteada por Edmonds en los años sesenta, es si existe o no un “buen" algoritmo para el problema del viajante. Hasta el día de hoy esta cuestión no ha sido resuelta. De hecho, el Instituto Clay de Matemáticas ha ofrecido una recompensa de un millón de dólares al que descubra un algoritmo eficiente para el TSP o demuestre la no existencia del mismo.

Por esta razón, el problema del viajante ha alcanzado una importante posición en la teoría de la complejidad. Según esta teoría, los problemas para los que existe un buen algoritmo de resolución son clasificados como , de tiempo polinómico. Por el contrario, los problemas para los que aún no se ha encontrado un algoritmo eficiente son clasificados como NP, de tiempo polinómico no determinista. Un resultado muy importante en este campo procede de un artículo de S. Cook (1971). Cook demostró que muchos de los problemas denominados como “difíciles” son computacionalmente equivalentes, en el sentido de que un algoritmo polinómico para uno de ellos puede utilizarse para resolver los demás en un tiempo también polinómico. Los problemas que se engloban en este grupo reciben el nombre de NP-duro. El método utilizado por Cook ha servido de inspiración para muchos otros autores, que han catalogado así cientos de problemas dentro de esta categoría.

Así, una de las principales cuestiones de la teoría de la complejidad es si existe o no un algoritmo de tiempo polinómico para cualquier problema NP-duro. De existir, entonces todos los problemas dentro de la categoría NP podrán resolverse en un tiempo polinómico y se concluirá que las clases P y NP son iguales. Este es uno de los problemas más destacados de matemáticas y es en este sentido que el Instituto Clay ha ofrecido un millón de dólares.

Cada año se presentan múltiples propuestas intentando demostrar que $P = NP$, normalmente proporcionando algún algoritmo eficiente para el TSP, lo que resulta lógico, pues el Problema del Agente Viajero es probablemente el más estudiado de todos los problemas clasificados como NP-duro. Hasta el momento, muchas de estas propuestas siguen sin estudiarse, pues muchos investigadores se inclinan más por la teoría de que $P \neq NP$. Esta cuestión ha hecho de la teoría de la complejidad un campo mucho más activo.

A pesar de no haberse encontrado ningún algoritmo eficiente para el problema general del agente viajero, sí se han encontrado algoritmos para algunos casos particulares del mismo, incluso en algunos casos se ha logrado demostrar la no existencia de dichos algoritmos. Este es uno de los aspectos más sorprendentes de la teoría de la complejidad: problemas de aparente mayor dificultad son resueltos frente a otros de apariencia sencilla que no lo son. En ese sentido, encontramos dos casos especiales de TSP:

- TSP euclídeo: Se trata de un problema de agente viajero ficticio, en el sentido de que las ciudades se sitúan aleatoriamente en un cuadrado de tamaño prefijado según una distribución uniforme. Las distancias entre los puntos se calculan según la métrica euclídea. La importancia de este problema es que se ha podido demostrar que pertenece a la categoría NP-duro.

- TSP con matriz de distancia triangular: Se ha demostrado que los problemas del viajante con matriz de distancias triangulares ($c_{ij} = 0,\ si\ i > j$ pueden resolverse en un tiempo polinómico, a pesar de su gran parecido con el TSP general.

El primer caso particular de TSP cuya estructura permitió aplicar un algoritmo de tipo polinómico fue un problema de secuenciación de tareas. En los últimos años ha sido destacable la literatura rusa que se ha concentrado en la identificación de casos "sencillos" del Problema del Agente Viajero.

6 JUSTIFICACIÓN

El potencial de esta investigación se precisa por la justificación y viabilidad de la misma, a través de los siguientes aspectos:

- **Conveniencia.** Los resultados obtenidos por este estudio serán de mucho provecho por su utilidad práctica para resolver diversos problemas sociales modelables según el TSP.
- **Relevancia Social.** Por citar un par ejemplos, las dificultades de transporte y la construcción de un canal interoceánico son los mayores problemas por resolver en Nicaragua. Nuestro país, hoy día, carece de un banco de personal capacitado en el área de la investigación de operaciones tanto en el ámbito de la investigación como en el académico de las universidades.
- **Implicaciones Prácticas.** Todo el proceso de investigación y los resultados mismos tendrán implicaciones prácticas debido a que permitirá resolver diversos problemas reales del área de la investigación de operaciones, tales como dificultades o inconvenientes de transporte, asignación de recursos, líneas de espera, mantenimiento y reemplazo de equipo, etc.
- **Valor Teórico.** Esta investigación tiene una valía significativa por tratar uno de los problemas abiertos más importantes de la investigación de operaciones y más estudiado por diferentes investigadores de muchas naciones ya que no se ha podido encontar solución óptima. Por tanto implicará un valioso aporte al conocimiento científico.
- **Utilidad Metodológica.** Esta tarea de investigación tendrá mucha utilidad metodológica porque se piensa que los resultados obtenidos constituirán un valioso andamiaje para futuras investigaciones del TSP. Asimismo, a nivel universitario, ingenieros de la computación y sistemas, civiles, industriales y electrónicos de la universidad tendrán suficientes herramientas metodológicas tanto para la inserción laboral como para la solución de situaciones específicas.
- **Viabilidad o Factibilidad de Ejecución.** Las principales razones que motivan este estudio son la viabilidad o factibilidad de ejecución del mismo. La posibilidad de llevar a cabo esta investigación radica, en primer lugar, en la necesidad que tiene actualmente la Facultad de Ciencias e Ingeniería de la UNAN-Managua de desarrollar investigaciones de vital importancia para el país. Asimismo, el aspecto económico no representa ningún factor negativo para el desarrollo de este estudio. Finalmente, la UNAN cuenta con el apoyo de un Doctor graduado en esta área de investigación, y cuya participación en investigaciones de esta naturaleza ha sido muy notoria.

7 OBJETIVOS

7.1 Objetivo General

Diseñar Algoritmos Heurísticos y Metaheurísticos eficientes para resolver el Problema del Agente Viajero o TSP (Traveling Salesman Problem).

7.2 Objetivos Específicos

- Revisar el estado del arte para el Problema del Agente Viajero con los resultados de investigaciones que se han desarrollado al día de hoy.
- Describir la teoría matemática de convexidad para formular el TSP.
- Diseñar los algoritmos heurísticos del Vecino Más Cercano y Greedy Randomized Adaptive Search Procedures (GRASP) para resolver el TSP.
- Diseñar los algoritmos metaheurísticos de Tabu Search y Scatter Search para resolver el TSP.
- Implementar los algoritmos heurísticos y metaheurísticos desarrollados en lenguaje C++ del ambiente de programación Microsoft Visual Studio 2008, Solver de Excel y WinQSB, utilizando las instancias de Travel Salesman Problem Library (TSPLIB).
- Comparar la eficiencia de los algoritmos propuestos mediante la librería virtual TSPLIB.

8 ANTECEDENTES

Los orígenes del Problema del Agente Viajero no son claros, el primer indicio de éste fue a partir de 1832 en Alemania, en un libro titulado: El problema del viajero, cómo debe hacer para obtener éxito en sus negocios, dónde en el último capítulo se vislumbra la esencia del TSP cuando se comenta que con una elección apropiada del tour, se puede ganar mucho tiempo y que el aspecto más importante es cubrir tantas ciudades como sean posibles sin visitar una de ellas dos veces, pero no contenía ningún tratamiento matemático.

El problema fue definido como tal en 1800 por el matemático irlandés William Rowan Hamilton y por el matemático británico Thomas Kirkman.

En 1857 el matemático Hamilton inventó el Icosian Game. Este es un juego que tiene por objetivo dar con el recorrido de hamiltoniano por las aristas de un dodecaedro para visitar una y sólo una vez cada vértice y que el de llegada coincida con el de partida. Este desafío se comercializó como un tablero con agujeros en cada nodo del grafo del dodecaedro y luego, fue distribuido en Europa en diversos formatos. Era un rompecabezas de recreo con base en la búsqueda de un ciclo hamiltoniano.

La forma general del TSP parece haber sido estudiado por primera vez por los matemáticos durante la década de 1930 en Viena y la Universidad de Harvard, en particular por Karl Menger, que define el problema, considerando el Algoritmo de Fuerza Bruta obvio, y observa la falta de optimización del heurístico del Vecino más Cercano: Se trata de un problema NP-duro en la optimización combinatoria, importante en la investigación de operaciones y ciencias de la computación teórica.

Merrill Flood fue el responsable de divulgar el nombre del TSP. Le habló acerca de él a A. W. Tucker en 1937. Tucker le comentó que recordaba haberlo escuchado de boca de Hassler Whitney de la Universidad de Princeton, quién estaba estudiando la teoría de grafos, en especial el problema de los cuatro colores, pero no podía confirmar con certeza esta historia. De ser verídica, asegura que ocurrió en los años 1931-1932 ya que fue entonces cuando se hallaba terminando su tesis con Lefschetz.

John Williams incitó a Flood en 1948 a popularizar el TSP en la corporación RAND en Santa Mónica, motivado por el propósito de crear talentos intelectuales para modelos fuera de la teoría de juegos. No hay dudas de que la reputación y la autoridad de RAND, que rápidamente se convirtió en el centro intelectual de muchas de las investigaciones sobre esta teoría, amplificó la publicidad de Flood.

La aparición del artículo "Soluciones de un Problema del Agente Viajero de gran tamaño" de Dantzig, Fulkerson y Johnson en el Journal of the Operations Research Society of América fue uno de los principales eventos en la historia de la optimización, ya que esto conllevó al estudio de la combinatoria y los problemas de programación lineal: el problema del transporte y el de asignación. Dantzig desarrolló el método simplex para resolver problemas de programación lineal.

En 1953, existían códigos de implementación efectivos del Método Simplex en general y adaptaciones especiales para los casos del problema del transporte y de asignación. En 1954 los matemáticos Dantzig, Fulkerson y Johnson hicieron un método que resolvía el problema del horario de los buques. La solución llegó varios años después de que el modelo se volviera obsoleto, pero logró sobrevivir porque el método podía ser usado para estudiar preguntas básicas de la teoría combinatoria.

Ford y Fulkerson escribieron su primer reporte sobre sistemas de flujo en 1956, iniciando de este modo un tópico mayor del cual derivó un resultado de Johnson sobre la secuencia de trabajos. El TSP. Sin embargo, no estaba relacionado a simple vista con estos desarrollos pero había esperanza de que lo estuviera.

Los tres matemáticos antes mencionados especulaban que, empezando de un tour óptimo o tal vez cercano al óptimo, era posible probar optimalidad utilizando pocas ecuaciones adicionales (llamadas cortes). El método parecía funcionar en pequeños problemas, por eso, pasaron a trabajar sobre uno de 49 ciudades.

Ellos sugirieron la posibilidad de que fuesen necesarios un gran número de cortes. Dantzig, optimista, le apostó a Fulkerson que el número de cortes necesarios eran a lo sumo 25, en cambio Fulkerson más pesimista opinaba que se necesitaban al menos 26. Las predicciones fueron bastante acertadas: la cantidad correcta resultó ser 26 pero en el papel que se publicó se decía que sólo 25 eran necesarios.

Dantzig, Fulkerson y Johnson no sólo resolvieron un TSP de tamaño considerable sino que también demostraron que la complejidad de la estructura de un problema de optimización combinatoria no era un obstáculo insuperable para resolverlo. Ellos utilizaron por primera vez el concepto de Branch and Bound, que es un método computacional muy popular, en particular, cuando se requiere que sólo algunas de las variables sean enteras.

Otro método que se ha aplicado en la resolución del TSP fue la programación dinámica pero debido a la enorme cantidad de condiciones que incluye puede resolver instancias de problemas relativamente pequeños.

Richard M. Karp mostró en 1972 que el problema del ciclo de Hamilton era NP-completo, que implica la NP-dureza de TSP. Esto suministra una explicación matemática de la dificultad computacional aparente de encontrar rutas óptimas.

Se hizo un gran avance en la década de 1970 y 1980, cuando Grotschel, Padberg, Rinaldi y otros lograron resolver exactamente los casos con un máximo de 2392 ciudades. En la década de 1990, Applegate, Bixby, Chvátal, y Cocine desarrollaron el programa Concorde que se ha utilizado en muchas soluciones de registro recientes.

Gerhard Reinelt publicó en 1991 el TSPLIB que contiene la descripción de una biblioteca para el Problema del Agente Aiajero que está destinada a proporcionar a los investigadores un amplio conjunto de problemas de prueba de diversas fuentes y propiedades. Es decir, una colección de instancias de referencias de diferentes dificultades, que ha sido utilizado por muchos grupos de investigación para la comparación de resultados, como por ejemplo, en 2006, Cook y otros calculan un recorrido óptimo a través de una instancia de 85900 ciudades dada por un problema de diseño microchip.

9 MARCO TEÓRICO

Los fundamentos matemáticos necesarios para la comprensión y solución del Problema del Agente Viajero, están organizados en apartados que obdecen a la siguiente estructura:

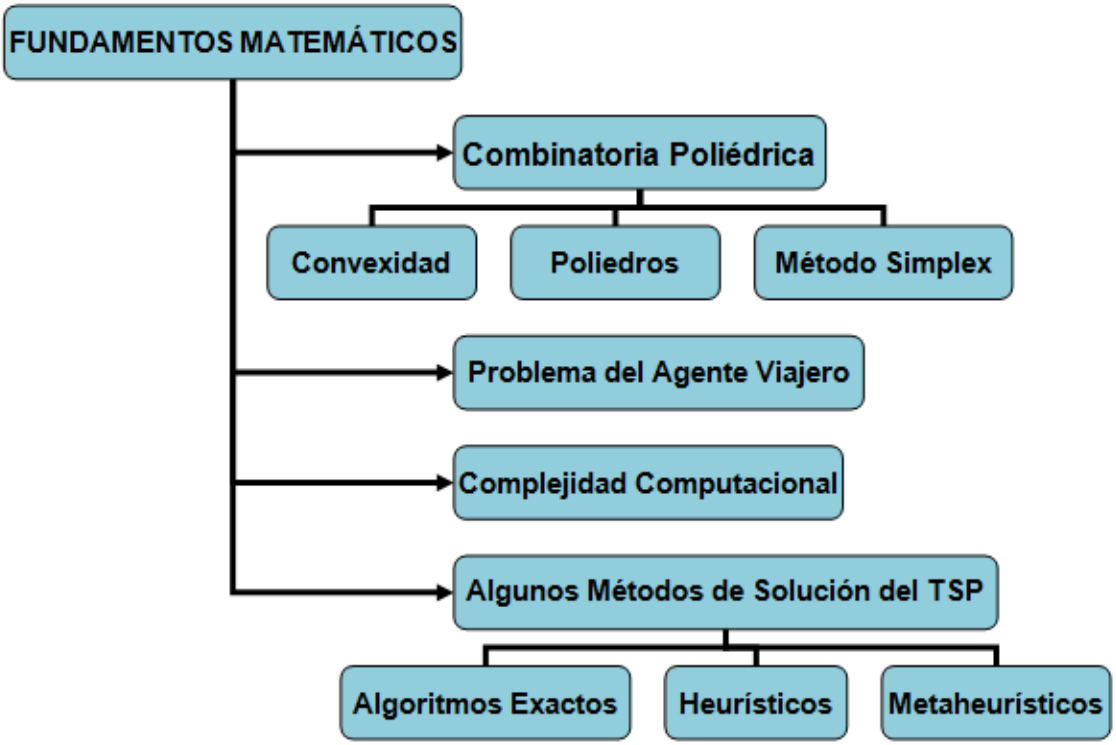


Figura 6: Estrcutura del Marco Teórico

9.1 Algunos Resultados de Combinatoria Poliédrica

9.1.1 Convexidad

El motivo de estudiar los conjuntos convexos radica en que tienen las mismas propiedades que los conjuntos soluciones de los problemas de programación lineal.

9.1.1.1 Algunas Definiciones Útiles

Definición 1 ***Combinación Lineal***

$x \in \mathbb{R}^n$ *es una combinación lineal de* $\{x_j\}_{j=1}^k \in \mathbb{R}^n \Leftrightarrow \exists \lambda_j \in \mathbb{R} : x = \sum_{j=1}^{k} \lambda_j x_j$

Definición 2 ***Combinación Cónica***

$x \in \mathbb{R}^n$ *es una combinación cónica de* $\{x_j\}_{j=1}^k \in \mathbb{R}^n \Leftrightarrow \exists \lambda_{j\geq 0} \in \mathbb{R} : x = \sum_{j=1}^{k} \lambda_j x_j$

Definición 3 ***Combinación Afín***

$x \in \mathbb{R}^n$ *es una combinación afín de* $\{x_j\}_{j=1}^k \in \mathbb{R}^n \Leftrightarrow \exists \lambda_j \in \mathbb{R}, \sum_{j=1}^{k} \lambda_j = 1 : x = \sum_{j=1}^{k} \lambda_j x_j$

Definición 4 ***Combinación Convexa***

$x \in \mathbb{R}^n$ *es una combinación convexa de* $\{x_j\}_{j=1}^k \in \mathbb{R}^n \Leftrightarrow \exists \lambda_{j\geq 0} \in \mathbb{R}, \sum_{j=1}^{k} \lambda_j = 1 : x = \sum_{j=1}^{k} \lambda_j x_j$

Definición 5 ***Independencia Lineal***

$\{x_j\}_{j=1}^k \in \mathbb{R}^n$ *son linealmente independiente* $\Leftrightarrow$ *la única solución de* $\sum_{j=1}^{k} \lambda_j x_j = 0$, *es* $\lambda_j = 0$, $j = 1, 2, ..., k, \forall \lambda_j \in \mathbb{R}$

- Se dice que el conjunto de vectores es linelamente independiente no que el vector es linealmente independiente.
- $\lambda_j = 0$ es condición necesaria pero no suficiente dado que $\lambda_j = 0$ no decide la independencia lineal, ya que esta solución siempre es posible por ser la trivial
- Un conjunto de vectores es LI o LD pero no ambas cosas a la vez.
- Un conjunto de vectores unitarios $\{e_1, e_2, ..., e_n\}$, es LI.
- $\{x_j\}_{j=1}^k \in \mathbb{R}^n$ es un conjunto de vectores linealmente dependiente si existe un vector del conjunto que se puede expresar como combinación lineal de los demás vectores, es decir $\{x_j\}_{j=1}^k \in \mathbb{R}^n$ es LD $\Leftrightarrow \exists x_i \in \{x_j\}_{j=1}^k : x = \sum_{j=1}^{k-1} \lambda_j x_j$
- Todo conjunto de vectores que contenga el vector nulo es LD. $C = \{x_1, x_2, ..., \vec{O}, ...x_n\}$ es LD ya que $\vec{O} \in C$

9.1.1.2 Algoritmo para Probar si un Conjunto es LI o LD

Definición 6 ***Independencia Afín***

$\{x_j\}_{j=1}^k \in \mathbb{R}^n$ *son afínmente independiente* $\Leftrightarrow$ *la única solución de* $\sum_{j=1}^{k} \lambda_j x_j = 0$, $\sum_{j=1}^{k} \lambda_j = 0$, $j = 1, 2, ..., k$, $\forall \lambda \in \mathbb{R}$.

$$\begin{cases} \sum_{j=1}^{k} \lambda_j x_j = 0 \\ \sum_{j=1}^{k} \lambda_j = 0 \end{cases} \Rightarrow \text{la solución es } \lambda_j = 0,\ j = 1, 2, ..., k$$

Nota: Independencia lineal $\Rightarrow$ Independencia Afín
Independencia afín $\nRightarrow$ Independencia lineal.

9.1.1.3 Algoritmo para Resolver Sistemas de Ecuaciones

- Si el determinante del sistema $\sum_{j=1}^{k} \lambda_j x_j = 0$ es $\neq 0 \Rightarrow$ solución única.
- Si el determinante del sistema $\sum_{j=1}^{k} \lambda_j x_j = 0$ es $0 \Rightarrow$ infinitas soluciones.
- El número máximo de vectores LI en $\mathbb{R}^n$ es n.
- El número máximo de vectores AI en $\mathbb{R}^{n+1}$ es $n + 1$.
- A_{mxn}, El rango de A, r(A) es el número máximo de filas (columnas LI).
- La dimensión de un espacio vectorial viene dado por el número de vectores de una base cualquiera de dicho espacio.
- En $\mathbb{R}^n$ la cantidad de vectores LI de un subcojunto de $\mathbb{R}^n$ es $\leq n$.

9.1.1.4 Los Tres Resultados Siguientes son Equivalentes:

1. $\{x_j\}_{j=1}^{k} \in \mathbb{R}^n$ son AI. $x_1 = [\ \ ]$, $x_2 = [\ \ ]$,..., $x_k = [\ \ ]$
2. $\{x_j - x_1\}_{j=2}^{k}$ son LI. $x_2 - x_1 = [\ \ ]$, $x_3 - x_1 = [\ \ ]$,..., $x_k - x_1 = [\ \ ]$
3. $\{(x_j, -1)\}_{j=1}^{k}$ son LI. $(x_1, -1) = \begin{bmatrix} x_1 \\ -1 \end{bmatrix}$, $(x_2, -1) = \begin{bmatrix} x_2 \\ -1 \end{bmatrix}$,..., $(x_k, -1) = \begin{bmatrix} x_k \\ -1 \end{bmatrix}$

Demostración $(a \Rightarrow b)$
$\{x_j\}_{j=1}^{k} \in \mathbb{R}^n$ son AI $\Rightarrow \{(x_j, x_1)\}_{j=2}^{k} \in \mathbb{R}^n$ son LI.

Queremos probar que

$$\sum_{j=1}^{k} \alpha_j (x_j - x_1) = \vec{0}$$

con $\alpha_j = 0$ y únicamente cero, $j = 1, 2, ..., k$

Sea $\sum_{j=1}^{k} \lambda_j (x_j - x_1) = \vec{0} \Rightarrow \sum_{j=1}^{k} \lambda_j x_j - \sum_{j=1}^{k} \lambda_j x_1 = \vec{0}$

Como los $\{x_j\}_{j=1}^{k}$ son AI, entonces

$$\begin{cases} \sum_{j=1}^{k} \alpha_j x_j = \vec{0} \\ \sum_{j=1}^{k} \alpha_j = 0 \end{cases}, \quad \alpha_j = 0, j = 1, 2, ...k$$

Sea $\lambda_j = \alpha_j, \quad j = 2, 3, ..., k$

$$\lambda_1 = -\sum_{j=2}^{k} \alpha_j, entonces \lambda_1 + \lambda_2 + ... + \lambda_k = \lambda_1 + \sum_{j=2}^{k} \lambda_j - \sum_{j=1}^{k} \lambda_j = 0$$

Y como $\alpha_j = 0, \quad j = 1, 2, ..., k$, entonces

$$\lambda_1 = 0 \quad y \quad \sum_{j=2}^{k} \lambda_j = 0$$

$\therefore \{x_j - x_1\}_{j=2}^{k}$ son LI.

Demostración $(b \Rightarrow c)$
$\{(x_j - x_1)\}_{j=2}^{k} \in \mathbb{R}^n$ son LI $\Rightarrow \{(x_j, -1)\}_{j=1}^{k} \in \mathbb{R}^{n+1}$ son LI.

Por notación $\{(x_j, -1)\}_{j=1}^{k} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_k \\ -1 \end{bmatrix} = \begin{bmatrix} x_j \\ -1 \end{bmatrix}$

Queremos probar que

$$\sum_{j=1}^{k} \lambda_j \begin{bmatrix} x_j \\ -1 \end{bmatrix} = \vec{0}_{n+1}, con \lambda_j = 0, \quad j = 1, 2, ..., k, \forall \lambda_j \in \mathbb{R}$$

$$\sum_{j=1}^{k} \lambda_j (x_j, -1) = \sum_{j=1}^{k} \lambda_j \begin{bmatrix} x_j \\ -1 \end{bmatrix} = \sum_{j=1}^{k} \lambda_j x_j - \sum_{j=1}^{k} \lambda_j = \begin{bmatrix} \vec{0_n} \\ 0 \end{bmatrix}_{\vec{0}_{n+1}}$$

De aquí $\sum_{j=1}^{k} \lambda_j x_j = \vec{0}_n$

Por otro lado $\underbrace{\sum_{j=1}^{k} \lambda_j = 0}_{un\ escalar} \Rightarrow - \sum_{j=1}^{k} \lambda_j x_l = 0 x_l, \quad x_l \in \mathbb{R}^n$

$$\Rightarrow - \sum_{j=1}^{k} \lambda_j x_l = 0 \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix}$$

$$\Rightarrow - \sum_{j=1}^{k} \lambda_j x_l = \vec{0}_n$$

Luego $\begin{cases} \sum_{j=1}^{k} \lambda_j x_j = \vec{0}_n \\ - \sum_{j=1}^{k} \lambda_j x_l = \vec{0}_n \end{cases} \Rightarrow \sum_{j=1}^{k} \lambda_j x_j - \sum_{j=1}^{k} \lambda_j x_l = \vec{0}_n$

Sacando factor común desde $j = 2$,

$$\sum_{j=2}^{k} \lambda_j (x_j - x_l) = \vec{0}$$

Y por hipótesis

$$\{x_j - x_1\}_{j=2}^{k} \Rightarrow \lambda_j = 0, j = 2, ..., k, \forall \lambda_j \in \mathbb{R}$$

■

9.1.1.5 Conjuntos Convexos

Definición 7 ***Recta***
Dados $a, b \in \mathbb{R}^n$ definimos la recta que pasa por a y b como el siguiente conjunto:
$r(a,b) = \{\in \mathbb{R}^n : x = (1-\lambda)a + \lambda b, \lambda \in \mathbb{R}\}$

- $a, b \in \mathbb{R}^n$, la recta que pasa por a y b, es el conjunto

$$r(a,b) = \{x \in \mathbb{R}^n : x = (1-\lambda)a + \lambda b, \lambda \in \mathbb{R}\}$$

- $a, b \in \mathbb{R}^n$, el segmento cerrado de extremos a y b es el conjunto

$$r[a,b] = \{x \in \mathbb{R}^n, x = (1-\lambda)a + \lambda b, \lambda \in [0,1]\}$$

- $a, b \in \mathbb{R}^n$, el segmento abierto de extremos a y b es el conjunto

$$r(a,b) = \{x \in \mathbb{R}^n, x = (1-\lambda)a + \lambda b, \lambda \in (0,1)\}$$

- Si $\lambda \in (0,1]$ o $[0,1)$ tendremos los segmentos semiabiertos $(a,b]$ y $[a,b)$

Geométricamente

$$r(a,b), \lambda \in \mathbb{R} \quad r(a,b) = \{x \in \mathbb{R}^2 : x = (1-\lambda)a + \lambda b,\ \lambda \in \mathbb{R}\}$$
$$[a,b] = \{x \in \mathbb{R}^2 : x = (1-\lambda)a + \lambda b,\ \lambda \in [0,1]\}$$
$$(a,b) = \{x \in \mathbb{R}^2 : x = (1-\lambda)a + \lambda b,\ \lambda \in (0,1)\}$$

De igual forma se grafican los semiabiertos
Por convenio

$$[a,a] = \{a\}$$
$$[a,a) = (a,a] = (a,a) = \phi$$

Definición 8 ***Combinación Convexa***
Se dice que el punto $x \in \mathbb{R}^n$ es combinación convexa de $\{x_j\}_{j=1}^n \in \mathbb{R}^n \Leftrightarrow \exists\ \{\lambda_j\}_{j=1}^k \in \mathbb{R}$ escalares no negativos $(\lambda_j \geq 0, \forall j = 1,2,...,k)$ cumpliendo $\sum_{j=1}^{k} \lambda_j = 1$ de forma que $x = \sum_{j=1}^{k} \lambda_j x_j$

Nota: Los puntos de cualquier segmento son combinaciones convexas de los extremos del segmento.
$x \in [a, b] \Rightarrow x = \lambda a + (1 - \lambda)b$
$x = \lambda_1 a + \lambda_2 b$
Como $\lambda \in [0, 1] \Rightarrow \lambda \geq 0$ y $\lambda + (1 - \lambda) = 1$

Nota: Una combinacíon convexa no es más que una combinación lineal en la que los escalares son no negativos y suman 1

Definición 9 ***Conjuntos Convexos***
Un subconjunto de $\in \mathbb{R}^n$*se dice que es convexo si y solo si contiene cualquier combinación convexa de cualquier pareja de elementos del conjunto.*

- $S \subseteq \in \mathbb{R}^n$, es convexo $\Leftrightarrow \forall a, b \in S \Rightarrow [a, b] \subseteq S$, lo que significa que

$$a\lambda + (1 - \lambda)b \in S, \forall a, b \in S, \forall \lambda \in [0, 1]$$

- $\mathbb{R}^n$ es convexo y por convenio $\emptyset$ y $\{x\}$, $x \in \mathbb{R}^n$ también lo son.
- Desde el punto de vista de la programación lineal los conjuntos convexos más interesantes de $\mathbb{R}^n$ son los hiperplanos y los semiespacios asociados, ya que las restricciones de cualquier problema de programación lineal son los conjuntos de alguno de estos dos tipos.

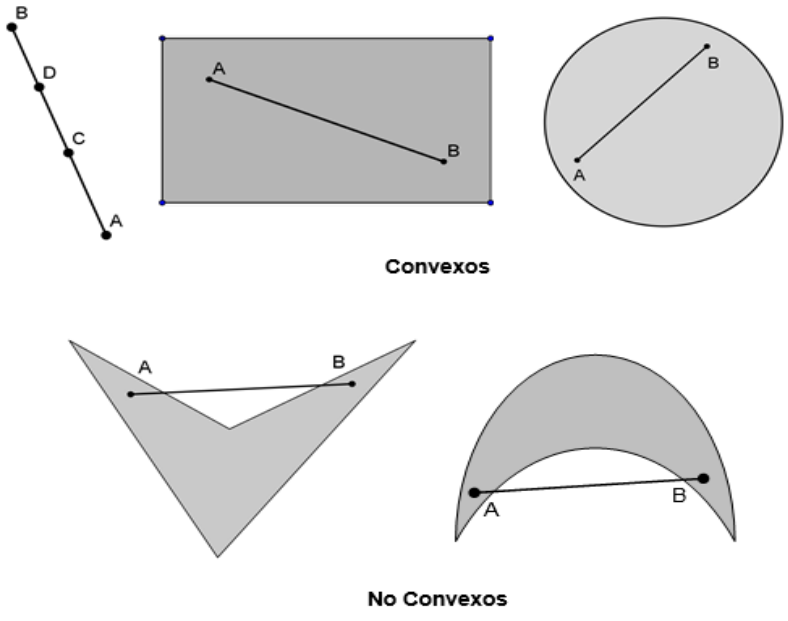


Figura 7: Conjuntos Convexos

Definición 10 ***Hiperplano***
Un hiperplano H en $\mathbb{R}^n$ *es un conjunto de la forma*
$H = \{x \in \mathbb{R}^n : a^t x = b\}$, $a \in \mathbb{R}^n \setminus \{\vec{0_n}\}$, $b \in \mathbb{R}$, *a es el vector normal al hiperplano.*

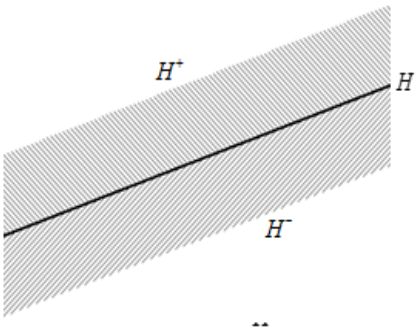


Figura 8: Hiperplano

Definición 11 ***Semiespacio***
Asociado a cualquier hiperplano se definen dos conjuntos más conocidos como semiespacios.
$H^+ = \{x \in \mathbb{R}^n : a^t x \geq b\}$ y $H^- = \{x \in \mathbb{R}^n : a^t x \leq b\}$

Proposición 12 *Los hiperplanos y los semiespacios son conjuntos convexos.*
$H = \{x \in \mathbb{R}^n : a^t x = b, b \in \mathbb{R}\}$, $a \in \mathbb{R}^n \setminus \{\vec{0}_n\}$, *es convexo.*

Demostración
Sean

$$y \in H \Rightarrow a^t y = b$$
$$z \in H \Rightarrow a^t z = b$$

Si H es convexo debe ocurrir que

$$\lambda y + (1-\lambda)z \in H \Rightarrow a^t[\lambda y + (1-\lambda)z] = b$$

$$a^t y = b \Rightarrow \lambda a^t y = \lambda b, \qquad \lambda y(1-\lambda) \geq 0$$
$$a^t z = b \Rightarrow \underline{(1-\lambda)a^t z = (1-\lambda)b}$$

Sumando miembro a miembro tenemos

$$\lambda a^t y + (1-\lambda)a^t z = \lambda b + (1-\lambda)b$$
$$a^t[\lambda y + (1-\lambda)z] = \lambda b + b - \lambda b$$
$$a^t[\lambda y + (1-\lambda)z] = b$$

$\therefore$ H es convexo
Las demostraciones para H^+ y H^- son análogas, solamente quitar el $=$ por el $\leq$ o $\geq$. ■

Proposición 13 *Los hiperplanos y los semispacios son conjuntos convexos.*
$H^- = \{x \in \mathbb{R}^n : a^t x \leq b, b \in \mathbb{R}\}, a \in \mathbb{R}^n \setminus \{\vec{0}\}, b \in \mathbb{R}$

Demostración
Sean $y, z \in H^- \Rightarrow a^t y \leq b \Rightarrow a^t z \leq b$
Debemos probar que

$$\begin{array}{l} \lambda y + (1-\lambda)z \in H^- \Rightarrow a^t[\lambda y + (1-\lambda)z] = b \\ a^t y \leq b \quad considerando \qquad \lambda \geq 0 \Rightarrow \lambda a^t y \leq \lambda b \\ a^t z \leq b \quad considerando \quad (1-\lambda) \geq 0 \Rightarrow \underline{(1-\lambda)a^t z \leq (1-\lambda)b} \end{array}$$

Sumado miembro a miembro tenemos:

$$\begin{array}{l} \lambda a^t y + (1-\lambda)a^t z \leq \lambda b + (1-\lambda)b \\ \lambda a^t y + (1-\lambda)a^t z \leq \lambda b + b - \lambda b \\ a^t[\lambda y + (1-\lambda)z] \leq b \end{array}$$

$\therefore H^-$ es convexo. ■

Proposición 14 *Dados $A, B \subseteq R^n$ dos conjuntos convexos, se cumple:*

1. *$\mu \in \mathbb{R} \Rightarrow \mu A$ es convexo.*
2. *$A + B$ es convexo.*
3. *$A - B$ es convexo.*
4. *La imagen y la anti-imagen por una aplicación lineal de un convexo es un convexo.*
   Sea $f : \mathbb{R}^n \to \mathbb{R}^n$ una función lineal.
   - *$f(A)$ es convexo $\forall A \subseteq \mathbb{R}^n$ convexo.*
   - *$f^{-1}(C)$ es convexo $\forall C \subseteq \mathbb{R}^m$ convexo.*

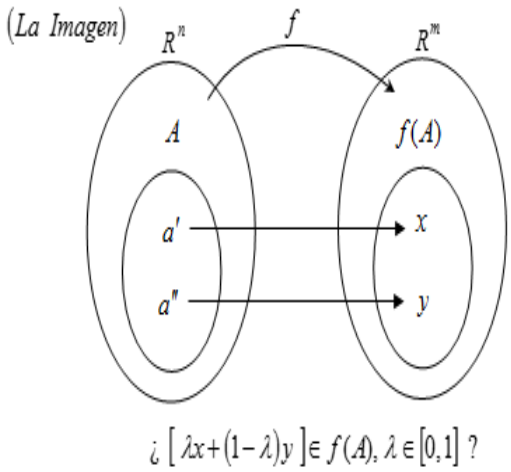


Figura 9: Imagen de un Convexo

Demostración 1

$A \subset \mathbb{R}^n$ convexo $\Rightarrow \mu A$ convexo, $\mu \in \mathbb{R}$

Sea

$$x \in \mu A \Rightarrow \exists a' \in A : x = \mu a'$$
$$y \in \mu A \Rightarrow \exists a'' \in A : y = \mu a''$$

Sea $\lambda \in [0, 1]$ queremos probar que

$$\lambda x + (1 - \lambda)y \in \mu A$$

Esto significa que

$$\lambda x + (1 - \lambda)y$$

Debe cumplir con la forma que tienen los elementos de μA, entonces

$$\lambda x + (1 - \lambda)y = \lambda\mu a' + (1 - \lambda)\mu a'' = \mu[\lambda a' + (1 - \lambda)a'']$$

Por ser A convexo

$$\lambda a' + (1 - \lambda)a'' \in A, \quad por \ \ cerradura.$$

Sea $a''' = \lambda a' + (1 - \lambda)a''$, por tanto $\mu a''' \in \mu A$.

Demostración 2

$A, B \subset \mathbb{R}^n$ Convexos $\Rightarrow A + B$ convexo.

Sea

$$x \in (A + B) \Rightarrow \exists a' \in A, b' \in B : x = a' + b'$$
$$y \in (A + B) \Rightarrow \exists a'' \in A, b'' \in B : y = a'' + b''$$

Sea $\lambda \in [0,1]$
Queremos probar que

$$\lambda x + (1-\lambda)y \in (A+B)$$

Debe tener la forma de los elementos de $(A+B)$, es decir una suma

$$\lambda x + (1-\lambda)y = \lambda(a'+b') + (1-\lambda)(a''+b'') = \lambda a' + \lambda b' + (1-\lambda)a'' + (1-\lambda)b''$$
$$= [\lambda a' + (1-\lambda)a''] + [\lambda b' + (1-\lambda)b'']$$

Sean

$$a''' = \lambda a' + (1-\lambda)a'' \in A \text{ y } b''' = \lambda b' + (1-\lambda)b'' \in B$$

Ya que A y B son convexos, luego

$$\lambda x + (1-\lambda)y = a''' \in A + b''' \in B \Rightarrow \lambda x + (1-\lambda)y \in (A+B)$$

Por tanto $(A+B)$ es convexo.

Demostración 3
$A, B \subset \mathbb{R}^n$ convexo $\Rightarrow A - B$ es convexo.
A es convexo, B convexo $\Rightarrow -B$ es convexo por que μA es convexo, $\mu \in \mathbb{R}$ prueba(1), luego $A + (-B)$ es convexo por prueba (2) por tanto $A + (-B) = A - B$ es convexo.

Demostración 4
La imagen por una aplicación lineal de un convexo es un convexo.
$f : \mathbb{R}^n \to \mathbb{R}^n, A \subset \mathbb{R}^n \to f(A)$ convexo?
Sea

$$x \in f(A) \Rightarrow \exists a' \in A : x = f(a')$$
$$y \in f(A) \Rightarrow \exists a'' \in A : y = f(a'')\lambda \in \mathbb{R}$$

Queremos probar que

$$\lambda x + (1-\lambda)y \in f(A)$$

Esto significa que

$$\exists a''' \in A : \lambda x + (1-\lambda)y = f(a''')$$
$$\lambda x + (1-\lambda)y = \lambda f(a') + (1-\lambda)f(a'') = f(\lambda a') + f[(1-\lambda)a''] = f[\lambda a' + (1-\lambda)a'']$$

Por ser f lineal.
Hacemos

$$a''' = \lambda a' + (1-\lambda)a'' \in A$$

Dado que A es convexo
Luego

$$\lambda x + (1-\lambda)y = f(a''')$$

$\therefore f(A)$ es convexo.
La anti-imagen por una aplicación lineal de un convexo es un convexo.
$C \in \mathbb{R}^n$ convexo $\Rightarrow f^{-1}(c)$ convexo? Sea

$$x \in f^{-1}(c) \Rightarrow \exists c' \in C : c' = f(x)$$
$$y \in f^{-1}(c) \Rightarrow \exists c'' \in C : c'' = f(y), \lambda \in [0,1]$$

Queremos probar que

$$\lambda x + (1-\lambda)y \in f^{-1}$$

Esto significa que

$$\exists c''' \in C : c''' = f[\lambda x + (1-\lambda)y]$$

Partimos de una combinación convexa en el conjunto de las imágenes

$$\lambda c' + (1-\lambda)c'' \in C$$

Por ser C un convexo

$$\lambda f(x) + (1-\lambda)f(y) = f(\lambda x) + f[(1-\lambda)y] = f[\lambda x + (1-\lambda)y] \in C$$

Finalmente tenemos

$$\underbrace{\lambda c' + (1-\lambda)c'' = c'''}_{por \quad ser \quad C \quad convexo} = f[\lambda x + (1-\lambda)y] \Rightarrow \lambda x + (1-\lambda)y \in f^{-1}(c)$$

■

Proposición 15 *Sea $\{A_i, i \in I\}$ una familia de convexos de $\mathbb{R}^n$, entonces $\bigcap_{i\in I} A_i$, es también un convexo.*

Demostración
Sean

$$\begin{cases} x, y \in \bigcap_{i\in I} A_i \\ \lambda \in [0,1] \end{cases}$$

Queremos probar que

$$(1-\lambda)x + \lambda y \in \bigcap_{i\in I} A_i$$
$$x, y \in \bigcap_{i\in I} A_i \Rightarrow x, y \in A_i, \forall i \in I$$

Como A_i es convexo

$$\forall i \in I \Rightarrow \lambda x + (1-\lambda)y \in A_i$$
$$\forall i \in I \Rightarrow \lambda x + (1-\lambda)y \in \bigcap_{i\in I} A_i$$

■

- La unión de convexos no es en general un convexo.

Definición 16 ***Envoltura Convexa***
Dado $S \subseteq \mathbb{R}^n$ se define la envoltura convexa de S, $Co(S)$ como el menor convexo en sentido de inclusión que contiene a S.

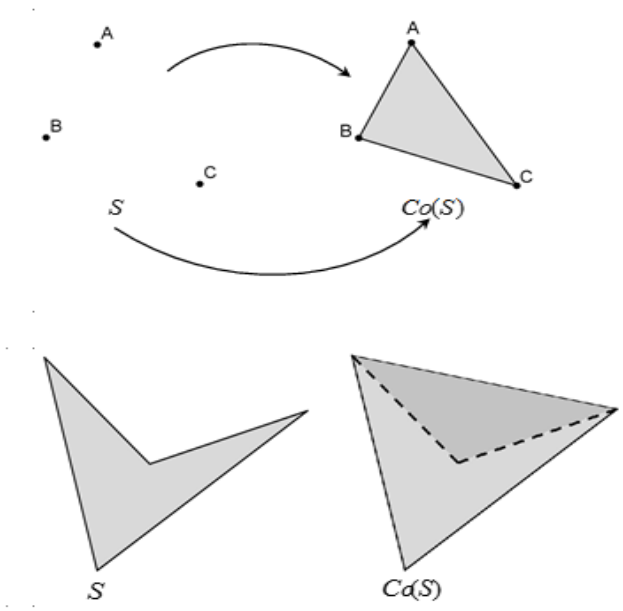


Figura 10: Envoltura Convexa

De la proposición y la definición anterior se deduce de forma inmediata que:

- La envoltura convexa de S es la intersección de todos los conjuntos convexos que contienen o son iguales a S $Co(S) = \left\{ \bigcap_{i \in I} \{A_i : A_i \quad convexo, S \subseteq A_i \right\}$
- S es convexo si coincide con su envoltura convexa. S es convexo $\Leftrightarrow S = Co(S)$
- La envoltura convexa de un número finito de puntos es un politopo.
- La envoltura convexa de $m + 1$ puntos afínmente independiente es un simplex de orden m.
- La independencia lineal $\Rightarrow$ la independencia afín.

Nota: $x_0, x_1, ..., x_m \in \mathbb{R}^n$ son afínmente independiente $\Leftrightarrow \{x_i - x_0\}_{i=1}^m$ es un sistema de m vectores linealmente independiente.

Una forma de caracterizar la independencia afín es:

$$\{x_0, x_1, ..., x_m\} \text{ son } A_i \Leftrightarrow \begin{cases} \sum_{j=0}^{m} \lambda_j x_j = 0 \\ \sum_{j=0}^{m} \lambda_j = 0 \end{cases} \Rightarrow \lambda_j = 0, \forall j$$

Lema 17 *Sea $S \subseteq \mathbb{R}^n$, entonces S es convexo $\Leftrightarrow$ S contiene todas las combinaciones convexas finitas de sus elementos.*

Vamos a generalizar la definición de conjuntos convexos (contiene cualquier segmento cuyos extremos están en el conjunto convexo)

Lema 18 *Sea $S \subseteq \mathbb{R}^n$, entonces S es convexo $\Leftrightarrow$ S contiene todas las combinaciones convexas finitas de sus elementos.*

Demostración $(\Leftarrow)$
Evidentemente, pues si S contiene cualquier combinación convexa de un número finito de elementos, en particular contendrá las combinaciones convexas de cualquier pareja de elementos.

Demostración $(\Rightarrow)$
Por inducción matemática.
$Para \quad k = 1$:
Dado

$$x_1 \in S y \lambda_1 \geq 0 \Rightarrow \lambda_1 x_1 \in S$$

Por ser S convexo y haciendo en particular $\lambda_1 = 1$. $\quad (1)$
$Para \quad k = 2$:
Dado

$$x_1, x_2 \in S \; y \; \lambda_1, \lambda_2 \geq 0 \in \mathbb{R}, \lambda_1 + \lambda_2 = 1$$

Por ser S convexo $\lambda_1 x_1 + \lambda_2 x_2 \in S$. $\quad (2)$
Suponiendo que es cierto para $\quad k - 1$:
Dado

$$\begin{cases} x_1, x_2, ..., x_{k-1} \in S \\ \lambda_1, \lambda_2, ..., \lambda_{k-1} \geq 0 \in \mathbb{R} \\ \lambda_1 + \lambda_2 + ... + \lambda_{k-1} = 1 \end{cases} \Rightarrow y = \sum_{j=1}^{k-1} \lambda_j x_j \in S, por ser S convexo. \quad (3)$$

Nos preguntamos ahora si se cumple para K.

Dado

$$\begin{cases} x_1, x_2, ..., x_k \in S \\ \lambda_1, \lambda_2, ..., \lambda_k \in S \lambda_j \geq 0, j = 1, 2, ..., k \\ \lambda_1 + \lambda_2 + ... + \lambda_k = 1 \end{cases} \Rightarrow y = \sum_{j=1}^{k} \lambda_j x_j \in S?$$

Dado Descomponemos y de la siguiente forma:

$$y = \sum_{j=1}^{k-1} \lambda_j x_j + \lambda_k x_k, \; donde \; A = \sum_{j=1}^{k-1} \; y \; B = \lambda_k x_k \quad (4)$$

$B \in S$ dado que por el punto (1) $\lambda_k \geq 0$ y $x_k \in S$ con S convexo, entonces

$$B = \lambda_k x_k \in S$$

Sobre A no tenemos garantía que pertenece a S. No lo podemos justificar con (3) porque nadie garantiza que

$$\sum_{j=1}^{k-1} \lambda_j = 1$$

Buscaremos la forma de que el primer sumando sea un punto de S y si lo logramos aplicamos (2). Realmente la prueba recae en darle la prueba de (3) que es cierto por lo supuesto anteriormente.
Hacemos

$$\mu = \lambda_1 + \lambda_2 + ... + \lambda_{k-1} = \sum_{j=1}^{k-1} \lambda_j$$

Como los $\lambda_j \geq 0$, $j = 1, ..., k$, entonces suponemos que $\mu \neq 0$, ya que si $\mu = 0$, entonces

$$\sum_{j=1}^{k} \lambda_j = \sum_{j=1}^{k-1} \lambda_j + \lambda_k = 0 + \lambda_k$$

Luego

$$\sum_{j=1}^{k} \lambda_j = \lambda_k = 1 \Rightarrow y = \lambda_k x_k = x_k$$

Y no habría nada que probar. Por tanto, tenemos que $\mu > 0$.
Sean

$$\mu_j = \tfrac{\lambda_j}{\mu}, j = 1, 2, ..., k-1, x = \sum_{j=1}^{k-1} \mu_j x_j \quad y \quad \sum_{j=1}^{k-1} \mu_j = \sum_{j=1}^{k-1} \frac{\lambda_j}{\mu} = \frac{1}{\mu} \sum_{j=1}^{k-1} \lambda_j = \frac{1}{\mu}(\mu) = 1$$

Por tanto hemos construido un punto x como una combinación lineal finita de puntos de S que es un convexo y por lo tanto pertenece a S.

$$\begin{cases} x = \sum_{j=1}^{k-1} \mu_j x_j, x_j \in S \\ \sum_{j=1}^{k-1} \mu_j = 1, \mu_j \geq 0 \end{cases} \Rightarrow x \in S$$

Ahora hacemos

$$x = \sum_{j=1}^{k-1} \mu_j x_j = x = \sum_{j=1}^{k-1} \frac{\lambda_j}{\mu} x_j = \frac{1}{\mu} x = \sum_{j=1}^{k-1} \lambda_j x_j \Rightarrow \mu x = \sum_{j=1}^{k-1} \lambda_j x_j$$

Sea $A = \mu x \in S$ por (1) se justifica que $\mu x \in S$, luego regresando a (4)

$$y = A + B \quad con \quad B \in S \quad y \quad A \in S \Rightarrow y = \mu x + \lambda_k x_k$$

Y hacemos

$$\lambda_k = 1 - \mu \Rightarrow y = \mu x + (1 - \mu) x_k \in S$$

Finalmente

$$y = \mu x + (1 - \mu) x_k = \sum_{j=1}^{k-1} \lambda_j x_j + \lambda_k x_k = \sum_{j=1}^{k} \lambda_j x_j$$

Por tanto

$$\mu + (1 - \mu) = \sum_{j=1}^{k-1} \lambda_j + \left(1 - \sum_{j=1}^{k-1} \lambda_j\right) = 1$$

■

Lema 19 *Sea* $S \subseteq \mathbb{R}^n \Rightarrow Co(S) = \{\sum_{j \in I} \lambda_j x_j : x_j \in S, \lambda_j \geq 0, \forall j \in I, \sum_{j \in I} \lambda_j = 1, |I| < \infty\}$.

Demostración

$$C = Co(S) = \left\{\sum_{j \in I} \lambda_j x_j : x_j \in S, \lambda_j \geq 0, \forall j \in I, \sum_{j \in I} \lambda_j = 1, |I| < \infty\right\}$$

Demostraremos que C es un convexo que contiene a S, contenido a su vez en $Co(S)$ y como $Co(S)$ es el menor conjunto convexo que contiene a S, tenemos el resultado perseguido i.e $C = Co(S)$

- Como $C = \left\{\sum_{j \in I} \lambda_j x_j : x_j \in S, \lambda_j \geq 0, \forall j \in I, \sum_{j \in I} \lambda_j = 1, |I| < \infty\right\}$, contiene todas las combinaciones finitas de elementos de S, es evidente que $S \subset C$.

- Se deduce del lema anterior y del hecho de que C es el conjunto de todas la combinaciones convexas de sus elementos.
- C es convexo.

Sean $x, y \in Co(S) \lambda \in [0,1] \Rightarrow z = \lambda x + (1-\lambda)y \in Co(S)$?
Como

$$x \in Co(S) \Rightarrow \exists J : |J| < \infty : x = \sum_{j \in J} \lambda_j x_j : x_j \in S, \lambda_j \geq 0, \forall j \in J, \sum_{j \in J} \lambda_j = 1$$
$$y \in Co(S) \Rightarrow \exists I : |I| < \infty : y = \sum_{i \in I} \mu_i x_i, \mu_i \geq 0, \forall i \in I, \sum_{i \in I} \mu_i = 1$$

Luego

$$z = \lambda x + (1-\lambda)y \Rightarrow z = \lambda \sum_{j \in J} \lambda_j x_j + (1-\lambda) \sum_{i \in I} \mu_i y_i$$
$$\Rightarrow z = \sum_{j \in J} \lambda \lambda_j x_j + \sum_{i \in I} (1-\lambda) \mu_i y_i$$

Finalmente

$$\sum_{j \in J} \lambda \lambda_j + \sum_{i \in I} (1-\lambda)\mu_i = \lambda \sum_{j \in J} \lambda_j + (1-\lambda) \sum_{i \in I} \mu_i = \lambda(1) + (1-\lambda)(1) = 1$$

■

Resumen
$Co(S)$ es el conjunto de todas las combinaciones finitas de puntos de S.
Dado $z = \lambda x + (1-\lambda)y$ se debe probar que $z \in Co(S)$, como x e $y \in Co(S)$ y $\lambda_1 = \lambda$, $\lambda_2 = (1-\lambda)$, basta probar que $\lambda_1 + \lambda_2 = 1$, partiendo de sustituir la definición de x e y en z.

El siguiente teorema nos permite dar un paso más y definitivo a la hora de caracterizar las envolturas convexas.

Teorema 20 ***Teorema de Caratheodory***
Sea $S \subseteq \mathbb{R}^n$ arbitrario. Si $x \in Co(S)$, entonces $\exists \{x_j\}_{j)1}^{n+1} \in S$ tal que $x \in Co(x_1, x_2, ..., x_n, x_{n+1})$, es decir

$$x = \sum_{j=1}^{n+1} \lambda_j x_j, \lambda_j \geq 0, \quad \forall j = 1, ..., n+1, \quad \sum_{j=1}^{n+1} \lambda_j = 1$$

Demostración

Se debe probar que, $\begin{cases} S \subseteq \mathbb{R}^n \\ S \; arbitrario \\ x \in Co(S) \end{cases} \Rightarrow \begin{cases} \exists \{x_j\}_{j=1}^{n+1} \in S, x_j \in S, \forall j = 1, ..., (n+1) \\ x \in Co(\{x_j\}_{j=1}^{n+1}), \lambda_j \geq 0, \sum_{j=1}^{n+1} \lambda_j = 1, \forall j \end{cases}$

- Si $x \in Co(S)$ entonces x se expresa como combinación convexa de un número finito de elementos de S, supongamos que son k puntos, si $k \leq n+1$, entonces no habría nada que probar.

- Supongamos que $k > (n+1)$, entonces probaremos que podemos quitar al menos un x_j de la combiandción inicial hasta llegar a $n+1$puntos.

- Como hemos asumido que $k > (n+1) \Rightarrow (k-1) > n$, con "$n$" la dimensión del espacio en que trabajamos.

- Con $(k-1) > n$ construimos un conjunto de $k-1$ vectores linealmentes dependientes, sea este $\{x_2 - x_1, x_3 - x_2, ..., x_k - x_1\} = \{x_j - x_1\}_{j=2}^{k}$.
  Es decir $\{x_j - x_1\}_{j=2}^{k}$ es LD $\Rightarrow \exists \mu_j \geq 0, \mu_j \in \mathbb{R}, j = 2, ..., k$, μ_j no todos nulos tal que $\sum_{j=2}^{k} \mu_j(x_j - x_1) = 0$

- Sea $\mu_1 = (\mu_2 + \mu_3 + \cdots + \mu_k = -\sum_{j=2}^{k} \mu_j = 0 \Rightarrow \sum_{j=2}^{k} \mu_j = 0$

- Aplicando distributividad en $\sum_{j=2}^{k} \mu_j(x_j - x_1) = 0$
  $\sum_{j=2}^{k} \mu_j x_j - \sum_{j=2}^{k} \mu_j x_1 = \sum_{j=2}^{k} \mu_j x_j - x_1 \sum_{j=2}^{k} \mu_j = \sum_{j=2}^{k} \mu_j x_j - x_1(-\mu_1)$
  $=\sum_{j=2}^{k} \mu_j x_j + x_1(\mu_1) = \sum_{j=2}^{k} \mu_j x_j = 0$

- De lo anterior obtenemos

$$\begin{cases} \sum_{j=1}^{k} \mu_j x_j = 0, x_j \in S, \forall j \\ \sum_{j=1}^{k} \mu_j = 0, \lambda_j \in \mathbb{R} \end{cases}$$

- Como los $\mu_j \geq 0$, supongamos que hay al menos uno positivo (sin pérdida de generalidad), lo que permitirá hacer los cocientes $\frac{\lambda_j}{\mu_j}$,$\mu_j \neq 0$ y luego tomar el mínimo $\alpha = min_{1<j<k}\{\frac{\lambda_j}{\mu_j}, \mu_j > 0\} = \frac{\lambda_h}{\mu_h} > 0, h \in \{1, 2, 3, ..., k\}$

- Por hipótesis del Teorema de Caratheodory $x = \sum_{j=1}^{k} \lambda_j x_j$, transformando esta expresión
$\sum_{j=1}^{k} \lambda_j x_j - \alpha \cdot 0_n = \sum_{j=1}^{k} \lambda_j x_j - \alpha \sum_{j=1}^{k} \mu_j x_j = \sum_{j=1}^{k} (\lambda_j - \alpha\mu_j) x_j$, es decir
$x = (\lambda_1 - \alpha\mu_1)x_1 + (\lambda_2 - \alpha\mu_2)x_2 + \cdots + (\lambda_h - \alpha\mu_h)x_h + \cdots + (\lambda_k - \alpha\mu_k)x_{1k}$

Donde $\lambda_h - \alpha\mu_h = \lambda_h - (\frac{\lambda_h}{\mu_h})\mu_h = \lambda_h - \lambda_h = 0$, esto nos permite quitar de la combinación al vector x_h, luego:

$$x = \sum_{j \neq h}^{k} (\lambda_j - \alpha\mu_j) x_j = \sum_{j \neq h}^{k} \Theta_j x_j, con \Theta_j = \lambda_j - \alpha\lambda_j$$

De lo anterior x se expresa como combinación convexa de a lo sumo $(k-1)$ elementos de S. Ahora debemos probar dos cosas: $\sum_{j=1}^{k} \Theta_j = 1$ y $\Theta_j \geq 0, \forall j \neq h$

$$\sum_{j=1}^{k} \Theta_j = \sum_{j=1}^{k} (\lambda_j - \alpha\mu_j) = \sum_{j=1}^{k} \lambda_j - \alpha \sum_{j=1}^{k} \mu_j = \sum_{j=1}^{k} \lambda_j - \alpha(0) = \sum_{j=1}^{k} \lambda_j = 1$$

Finalmente $\Theta_j \geq 0$ $(\lambda_j - \alpha\mu_j) \geq 0$

Si $\mu_j = 0$, entonces $\lambda_j - \alpha\mu_j = \lambda_j$ y por hipótesis $\lambda_j \geq 0$

Si $\mu_j < 0$ $\lambda_j - \alpha\mu_j > 0$, luego $\lambda_j > \alpha\mu_j$ y $\Theta > 0$

Si $\mu_j > 0$, entonces sabemos que $\frac{\lambda_j}{\mu_j} \geq \frac{\lambda_h}{\mu_h}, \forall j \neq h$ con $\alpha = \frac{\lambda_h}{\mu_h}$.

Por tanto de la desigualdad anterior resulta que $\frac{\lambda_j}{\mu_j} \geq \alpha \Rightarrow \lambda_j \geq \alpha\mu_j$, de donde $\lambda_j - \alpha\mu_j \geq 0$. ■

Definición 21 ***Cono***
Un subconjunto $S \subseteq \mathbb{R}^n$ diremos que es un cono de vértice $\vec{0}_n$ si y sólo si $\forall x \geq 0$, es decir:

$$S \subseteq \mathbb{R}^n, \text{ es } k(\vec{0}_n) \Leftrightarrow \forall x \in S \Rightarrow \lambda x \in S, \forall \lambda \geq S$$

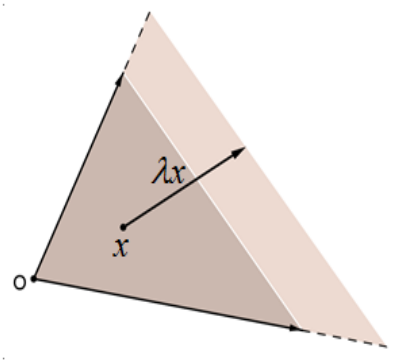


Figura 11: Cono

- Un cono convexo es un conjunto convexo que además es cono.
- $y \in \mathbb{R}^n$ es combinación no negativa (positiva) de $x_1, x_2, ..., x_k \in \mathbb{R}^n$ si y sólo si $\lambda_1, \lambda_2, ..., \lambda_k \in \mathbb{R}$, escalares no negativos que cumplen $y = \sum_{j=1}^{k} \lambda_j x_j$.
- Sea $\{A_i, i \in I\}$ una familia de conos convexos de $\mathbb{R}^n$, entonces $\bigcap_{i \in I} A_i$ es un cono convexo.
- $k(s) = \bigcap\{A_i : A_i \text{ es un cono convexo}, S \subseteq A_i\}$.
- S es un cono convexo si y solo si $S = k(S)$.
- Sea $S \subseteq \mathbb{R}^n$, S es un cono convexo si y sólo si S contiene todas las combinaciones positivas (no negativas finita de sus elementos).
- $S \subseteq \mathbb{R}^n \Rightarrow k(S) = \{\sum_{j \in I} \lambda_j x_j : x_j \in S, \lambda_j \geq 0, \forall j \in I < \infty\}$, envoltura cónica convexa.

Hemos presentado dos nuevos tipos de combinaciones lineales de elementos de $\mathbb{R}^n$ las:

- Combinaciones convexas $\rightarrow$ caso particular de combinación afín.
- Combinaciones positivas $\rightarrow$ caso particular de combinación lineal.

Asociadas a cada una de estas combinaciones tenemos las envolturas:

- Convexas $C_0(S)$ subconjunto de variedad afín $Af(S)$
- Cónico convexa $K(S)$ subconjunto de subespacio vectorial $Lin(S)$

Evidentemente

$$Co(S) \subseteq k(S) \subseteq lin(S)$$
$$Co(S) \subseteq Af(S) \subseteq lin(S)$$

En general ninguna de las otras inclusiones es cierta.

El objetivo del tema conjuntos convexos radica en que sus propiedades son la de los conjuntos de soluciones de los problemas de programación lineal.

Proposición 22 *Un subconjunto S de $\mathbb{R}^n$ que es un cono, es convexo si y solo si es cerrado para la suma, i.e. $\forall x, y \in S$ se cumple $(x + y) \in S$.*

$S \subset \mathbb{R}^n$, S un cono convexo $\Leftrightarrow \forall x, y \in S$, $x + y \in S$

Demostración $(\Rightarrow)$ Aquí probaremos que la suma está en S
Sea

$$x \in S \Rightarrow \lambda x = \tfrac{1}{2}x \in S \; por \; ser \; S \; un \; cono, \; \lambda \geq 0$$
$$y \in S \Rightarrow \lambda y = \tfrac{1}{2}y \in S \; por \; ser \; S \; un \; cono, \; \lambda \geq 0$$
$$z = \tfrac{1}{2}x + \tfrac{1}{2}y \in S \; por \; ser \; S \; un \; cono, \; \lambda \geq 0$$

luego

$$\lambda z = 2z \in S \; por \; ser \; S \; un \; cono, \; \lambda \geq 0$$
$$\lambda z = 2(\tfrac{1}{2}x + \tfrac{1}{2}y) = x + y$$

Demostración $(\Leftarrow)$ Aquí probaremos que el cono es convexo.
Sea

$$x, y \in S \Rightarrow \lambda x + (1 - \lambda)y \in S?$$
$$\lambda x \in S \; por \; ser \; S \; un \; cono, \; y \; \lambda \geq 0, \lambda \in [0, 1]$$
$$\lambda(1 - \lambda)x \in S \; por \; ser \; S \; un \; cono, \; y \; (1 - \lambda) \geq 0, \lambda \in [0, 1]$$

Como S es cerrado para la suma tenemos que $\lambda x + (1 - \lambda)y \in S$ ■

Definición 23 ***Envoltura Cónico Convexa.***
Dado $S \subseteq \mathbb{R}^n$ se define la envoltura cónico convexa de S, como el menor cono convexo que contiene a S. Se denota por $K(S)$.

Nota: La mayoría de resultados vistos para conjuntos convexos, admiten una versión en términos de conos convexos. Lo mismo sucede con las envolturas convexas y las cónico convexas.

9.1.2 Poliedros

9.1.2.1 Puntos Extremos, Direcciones Extremas y Poliedros

Definición 24 ***Punto Extremo***
Dado $S \subseteq \mathbb{R}^n$ un conjunto convexo no vacío. Sea x es un punto extremo de S si y solo si x no puede expresarse como combinación convexa estricta de dos puntos distintos de S.

Equivalentemente: $\forall x_1, x_2 \in S$, $\lambda \in (0,1)$, $x = \lambda x_1 + (1-\lambda)x_2$

- Se denota $ext(S)$ al conjunto de los puntos extremos de S
- Cuando S es compacto (cerrrado y acotado), cualquier punto de S se puede expresar como combinación convexa de sus puntos extremos, pero cuando no es acotado, no se puede decir lo mismo, por tanto hablamos de direcciones y direcciones extremas.

Definición 25 ***Dirección***
Un vector d de $\mathbb{R}^n$ no nulo $(d \neq \vec{o}_n)$ se dice que es una dirección de S siendo S un conjunto convexo si $\forall x \in S$, $(x + \lambda d) \in S, \lambda \geq 0$

$$\begin{cases} (d \neq \vec{o}_n) \in \mathbb{R}^n \\ S \subseteq \mathbb{R}^n, S \ es \ convexo \\ \forall x \in S (x + \lambda d) \in S, \forall \lambda \geq 0 \end{cases} \Rightarrow d \, es \, una \, \textbf{dirección} \, de \, \mathbf{S}.$$

- Se dice que dos direcciones d_1 y d_2 de S son distintas si y solo si $d_1 \neq \alpha d_2, \forall \alpha \geq 0$.

Definición 26 ***Dirección Extrema***
Se dice que una dirección S, es dirección extrema si y solo si no se puede obtener como una combinación positiva de dos direcciones de S, equivalentemente,

$$\forall d_1, d_2 \in S : d = \lambda_1 d_1 + \lambda_2 d_2, \, \lambda_1, \lambda_2 > 0 \Rightarrow d_1 = d_2 = d \equiv d_2 = \alpha d_1, \alpha > 0.$$

- Cualquier dirección de S se puede expresar como combinación positiva de d_1 y d_2

Definición 27 ***Poliedro***
Un conjunto $S \subseteq \mathbb{R}^n$ se dice que es un poliedro o conjunto poliédrico si es la intersección de un número finito de semi-espacios, i.e.

$$S = \bigcap_{i \in I} H_i^-, \quad \text{donde}$$
$$H_i^- = \{x \in \mathbb{R}^n / (a^i)^t x \leq b_i\}, i \in I = \{1, 2, ..., m\}$$

De forma equivalente
$(a^i)^t x \leq b_i$

$$(a^1)^t x \leq b_1 \Rightarrow [a_1^t a_2^t \cdots a_n^t] \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \leq b_1 \equiv \sum_{j=1}^{n} a_j^1 x_j \leq b_1$$

$$(a^2)^t x \leq b_2 \Rightarrow [a_1^2 a_2^2 \cdots a_n^2] \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \leq b_2 \equiv \sum_{j=1}^{n} a_j^2 x_j \leq b_2$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$(a^m)^t x \leq b_m \Rightarrow [a_1^m a_2^m \cdots a_n^m] \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \leq b_m \equiv \sum_{j=1}^{n} a_j^m x_j \leq b_m$$

- Si construimos una matriz cuyas filas son los vectores normales asociados a los hiperplanos que definen el poliedro, es posible representar el conjunto poliédrico como:

$$S = \{x \in \mathbb{R}^n : Ax \leq b\}, A_{mxn}, b \in \mathbb{R}^n$$

- Las desigualdades que pueden eliminarse sin modificar el poliedro se conocen como desigualdades redundantes.
- Un cojunto de puntos del poliedro que pertenece a uno o más de los hiperplanos que definen el poliedro se dice que es una cara del poliedro (vértice-arista).
- Un cono poliédrico es un poliedro formado por semi-espacios cuyos hiperplanos pasan por el origen.
- Un poliedro acotado es un politopo.
- Cualquier poliedro puede admitir, gracias a la existencia de restricciones redundantes, infinitas representaciones.

Proposición 28 *Dado* $S = \{x \in \mathbb{R}^n / Ax \leq b, x \geq 0\}$, A *una matriz* mxn, *un punto* $\bar{x}$ *es un punto extremo si y sólo si existen "n" hiperplanos en la descripción de* S, *que pasan por* $\bar{x}$ *y son linealmente independientes, es decir:*

Demostración $(\Rightarrow)$
Por Reducción Al Absurdo (RAA)

- Vamos a considerar que no son "n", sino que son "q" hiperplanos los que pasan por $\bar{x}$ con $q \neq n$. No vamos a considerar el caso $q > n$ porque los hiperplanos serían linealmente dependientes, por tanto consideramos $q < n$.

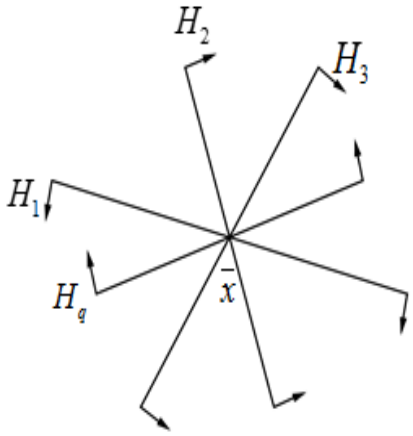


Figura 12: Hiperplanos

- Sea $Gx = g$ con G_{qxn}, x_{nx1},g_{qx1} el sistema lineal asociado a G. Donde G es submatriz de

$$\begin{bmatrix} A \\ \vdots \\ -I_n \end{bmatrix}$$

- El sistema lineal homogéneo asociado a G tiene al menos una solución no nula d talque $Gd = 0$, $d \neq 0$.

- También $\bar{x}$ es solución de $Gx = g$ ya que está contenido en los hiperplanos que componen G.

- De lo anterior podemos hacer

$$\begin{cases} a' = \bar{x} + \xi d \\ a'' = \bar{x} - \xi d \end{cases}$$

- Si x' y x'' estuvieran en S deben cumplir $Gx = g$, entonces:

$$\begin{aligned} x' = \bar{x} + \xi d \quad &\Rightarrow \quad Gx'g \\ &\Rightarrow \quad G(\bar{x} + \xi d) = g \\ &\Rightarrow \quad G\bar{x} + G\xi d = g \\ &\Rightarrow \quad G\bar{x} + \xi(Gd) = g \\ &\Rightarrow \quad G\bar{x} + \xi(0) = g \\ &\Rightarrow \quad G\bar{x} + 0 = g \\ &\Rightarrow \quad G\bar{x} = g \\ &\Rightarrow \quad x' \in S \\ &\Rightarrow \quad x'' = \bar{x} - \xi d \\ &\Rightarrow \quad Gx'' = g \\ &\Rightarrow \quad x'' \in S \end{aligned}$$

Luego sumando $x' = \bar{x} + \xi d$ con $x'' = \bar{x} - \xi d$:

$$\begin{array}{c} x' = \bar{x} + xid \\ x'' = \bar{x} - \xi \\ \hline x' + x'' = 2\bar{x} + \xi d - \xi d \Rightarrow \bar{x} = \frac{1}{2}x' + \frac{1}{2}x'' \text{ con } x' \neq x'' \end{array}$$

Contradicción ¡¡¡$\bar{x} = \frac{1}{2}x' + \frac{1}{2}x''$!!!, ya que $\bar{x}$ es un punto extremo y no se puede expresar como combinación convexa de dos puntos diferentes de S.

Por tanto, lo asumido es falso, lo cierto es lo que se quería demostrar, o sea, existen "n" hiperplanos en la descripción de S que pasan por $\bar{x}$ y son L.I.

Demostración $(\Leftarrow)$
Ahora probaremos que $\bar{x}$ es un punto extremo de S. Supongamos que $\bar{x} \in S$ y que se obtiene como solución de un Sistema de Ecuaciones Lineales (SEL) de "n" ecuaciones L.I. en $\mathbb{R}^n$, entonces $\bar{x}$ es solución única del sistema $Gx = g$, con G submatriz de

$$\begin{bmatrix} A \\ -I_n \end{bmatrix}$$

y subvector de

$$\begin{bmatrix} b \\ 0 \end{bmatrix}$$

Si $\bar{x}$ no fuese punto extremo de S, entonces existirían x' y $x'' \in S$ con $x' \neq x''$, $\lambda \in (0,1)$ talque $\bar{x} = (1-\lambda)x' + \lambda x''$.

Sea $\alpha x \leq \beta$ uno de los semiespacios en $Gx \leq g$, luego $\beta x = \alpha\bar{x} = \alpha[(1-\lambda)x' + \lambda x''] = \alpha(1-\lambda)x' + \alpha\lambda x'']$

Además, $\beta = \underbrace{(1-\lambda)}_{>0}\underbrace{\alpha}_{\leq\beta}x' + \underbrace{\lambda}_{>0}\underbrace{\alpha}_{\leq\beta}x'' \leq (1-\lambda)\beta + \lambda\beta \leq \beta - \lambda\beta + \lambda\beta \leq \beta$

Luego, $\beta \leq \beta$, pero $\beta \not< \beta$ sino que $\beta = \beta$ entonces

$$\left.\begin{aligned} \alpha\bar{x} &= \beta \\ \alpha x' &= \beta \\ \alpha x'' &= \beta \end{aligned}\right\} \quad x' = x'' = \bar{x}$$

$\therefore$ El SEL $Gx = g$ tiene solución única, luego $\bar{x}$ es un punto extremo. ■

9.1.2.2 Caracterización de Puntos Extremos y Direcciones Extremas de Poliedros

Definición 29 ***Punto Extremo Degenerado***
Un punto extremo es degenerado cuando el número de hiperplanos que pasan por él es superior a "n".

- En $\mathbb{R}^2$ los puntos extremos de los poliedros se obtienen como intersección de dos hiperplanos linealmente independientes. Los puntos extremos degenerados se obtendrán como intersección de 3 o más hiperplanos.
- En $\mathbb{R}^3$ los puntos extremos de los poliedros se obtienen como intersección de tres hiperplanos linealmente independientes. Los puntos extremos degenerados se obtendrán como intersección de 3 hiperplanos.
- La degeneración no necesariamente implica la existencia de desigualdades redundantes.

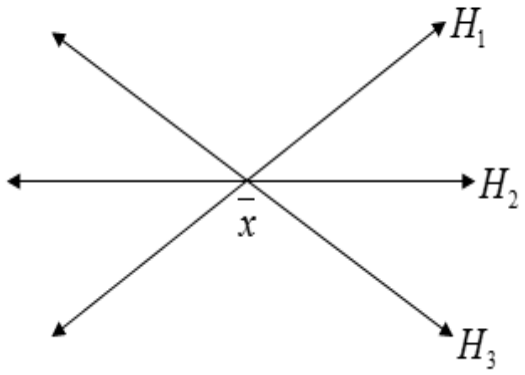


Figura 13: Punto Extremo Degenerado

- Sin pérdida de generalidad supondremos que estamos trabajando con un poliedro de la forma $S = \{x \in \mathbb{R}^n / Ax = b, x \geq \mathbf{0}_n\}$, A_{mxn}, $rg(A) = mb \in \mathbb{R}^m$. Tampoco hay pérdida de generalidad en suponer que es de rango completo por filas ya que si hubiese alguna restricción redundante podríamos eliminarla.

- El suponer $Ax = b$ permite claridad y simplicidad a las demostraciones de los resultados estudiados.

- Las columnas de A (y consecuentemente las componentes de x) han sido reorganizadas de forma que los "m" vectores sean L.I., i.e. una base de $\mathbb{R}^m$.

  $A_{mxn} \cdot x_n = b_n \quad \overrightarrow{reordenando} \quad \underbrace{B_{mxn} \cdot N_{mx(n-m)}}_{A} \begin{bmatrix} A \\ -I_n \end{bmatrix}$

  $Ax = b$

  $\left[B \vdots N\right] \cdot \begin{bmatrix} x_B \\ \dots \\ -x_n \end{bmatrix}$=b

  $B \cdot x_B + N \cdot x_N = b$ $\qquad$ B_{nxm} matriz de rango completo.

  $x_B \geq \mathbf{0}_m$, $x_N \geq \mathbf{0_{n-m}}$ $\qquad$ $N_{mx(n-m)}$ matriz no básica.

Teorema 30 ***(Teorema de Caracterización de Puntos Extremos)***
Dado $S = \{x \in \mathbb{R}^n / Ax = b, x \geq \mathbf{0}_n\}$ *un poliedro en donde* A *es una matriz* mxn *de rango completo y* $b \in \mathbb{R}^m$. $x \in S$ *es un punto extremo de* S *si y solo si es posible descomponer la matriz* A *en dos submatrices* $\left[B \vdots N\right] = A$ *de modo que*

$$\left[\begin{array}{c} x_B \\ x_N \end{array}\right] = \left[\begin{array}{c} B^{-1}b \\ \mathbf{0}_{n-m} \end{array}\right]$$

Donde B es una matriz no singular mxm y $B^{-1}b \geq \mathbf{0}_m$

Se debe probar que:

$$\left.\begin{array}{r} S = \{x \in \mathbb{R}^n / Ax = b, x \geq \mathbf{0}_n\} \\ A_{mxn}, b \in \mathbb{R}^m \\ x \in S, x \in extr(S) \end{array}\right\} \Leftrightarrow \left\{\begin{array}{l} \text{A=}\left[\text{B} \vdots \text{N}\right], x = \left[\begin{array}{c} x_B \\ x_N \end{array}\right] = \left[\begin{array}{c} B^{-1}b \\ \mathbf{0}_{n-m} \end{array}\right] \\ B_{mxm} / BB^{-1} = B^{-1}B = I, B^{-1}b \geq \mathbf{0}_m \end{array}\right.$$

Demostración $(\Leftarrow)$
Debemos probar tres cosas: primero que $x \geq \mathbf{0}_n$, segundo $x \in S$ y tercero $x \in extr(S)$.

a) $x \geq \mathbf{0}_n$?

Por hipótesis $\left[\begin{array}{c} x_B \\ x_N \end{array}\right] = \left[\begin{array}{c} B^{-1}b \\ \mathbf{0}_{n-m} \end{array}\right] \geq \left[\begin{array}{c} \mathbf{0}_m \\ \mathbf{0}_{n-m} \end{array}\right] \geq \mathbf{0}_n$

b) $x \in S$?

Esto significa que x satisface $Ax = b$

$$Ax = \left[B \vdots N\right] \left[\begin{array}{c} x_B \\ x_N \end{array}\right] = Bx_B + Nx_N = B(B^{-1}b) + N\mathbf{0}_{n-m} = B\bar{B}b - \mathbf{0}_m$$
$$Ax = Ib + \mathbf{0}_m = b$$
$$Ax = b$$

c) $x \in extr(S)$?

Sabemos que un punto extremo no puede ser representado como combinación convexa de dos puntos cualesquiera (diferentes) de S,

$$x \in extr(S) \Rightarrow \left\{\begin{array}{l} x \neq \lambda x_1 + (1-\lambda)x_2 \\ x = \lambda x_1 + (1-\lambda)x_2 \end{array}\right. \Rightarrow x = x_1 = x_2$$

Supongamos que lo podemos representar

$$x = \lambda x_1 + (1-\lambda)x_2$$
$$x = \lambda \begin{bmatrix} x_{1B} \\ x_{1N} \end{bmatrix} + (1-\lambda) \begin{bmatrix} x_{2B} \\ x_{2N} \end{bmatrix} = \begin{bmatrix} \lambda x_{1B} \\ \lambda x_{1N} \end{bmatrix} + \begin{bmatrix} (1-\lambda)x_{2B} \\ (1-\lambda)x_{2N} \end{bmatrix}$$

$$x = \begin{bmatrix} \lambda x_{1B} + (1-\lambda)x_{2B} \\ \lambda x_{1N} + (1-\lambda)x_{2N} \end{bmatrix}, pero \;\; x_{1N} = x_{2N} = 0 \;\; por\ hipótesis$$

$$x = \begin{bmatrix} \lambda x_{1B} + (1-\lambda)x_{2B} \\ \mathbf{0}_{n-m} \end{bmatrix}$$

Como $x_1, x_2 \in S$, entonces deben satisfacer $Ax = b$:

$$\begin{aligned} & x_1 \in S \Rightarrow Ax_1 = b \\ & \Rightarrow \left[B \vdots N\right] \begin{bmatrix} x_{1B} \\ x_{1N} \end{bmatrix} = b \\ & \Rightarrow Bx_{1B} + Nx_{1N} = b \\ & \Rightarrow Bx_{1B} = b \\ & \Rightarrow B^{-1}Bx_{1B} = B^{-1}b \\ & \Rightarrow Ix_{1B} = B^{-1}b \\ & \Rightarrow x_{1B} = B^{-1}b \end{aligned}$$

Análogamente, $x_{2B} = B^{-1}b$, por tanto:

$$\begin{gathered} x_{1B} = x_{2B} = x_B = B^{-1}b \Rightarrow x_{1B} = x_{2B} = x_B \\ \Downarrow \\ x_1 = x_2 = x \\ \Downarrow \\ x \in extr(S) \end{gathered}$$

Demostración $(\Rightarrow)$
La hipótesis la podemos resumir en lo siguiente: Si $x \in extr(S)$, entonces $x \in S$ y $x \geq \mathbf{0}_n$(Def. de Poliedro).

Como $x \geq \mathbf{0}_n$, supongamos sin pérdida de generalidad que las primeras k-componentes son positivas:

$$A = \left.\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_k \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right\} \begin{array}{l} k \text{ componentes positivas } x_j, \forall j = 1, 2, ..., k \\ \\ \\ \\ \\ \\ n - k \text{ componentes nulas } x_j = 0, \forall j = (k+1), ..., k \\ \\ \\ \end{array}$$

Queremos probar que los vectores columnas de A, asociados a las componentes $x_1, x_2, ..., x_k$ son linealmente independientes, o sea $Ax = b$.

Por reducción al absurdo supongamos que $\{a_j\}_{j=1}^{k}$ son L.D., entonces $\exists \lambda_j \in \mathbb{R}$ no todos nulos, tales que:

$$\sum_{j=1}^{k} \lambda_j a_j = \mathbf{0}_m$$

O sea,

$$\lambda_1 a_1 + +\lambda_2 a_2 + \ldots + \lambda_k a_k = \mathbf{0}_m \text{ con algunos } \lambda_j \neq 0$$

Con los λ_j construimos un vector dirección en $\mathbb{R}^n$:

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \vdots \\ \lambda_k \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{0}_m$$

Sean $x_1, x_2 \in S$ con $x_1 \neq x_2$, entonces

$$Ax_1 = A(x + \alpha\lambda) = Ax = \alpha \sum_{j=1}^{k} x_j a_j$$

De donde

$$Ax_1 = b + \alpha(\mathbf{0}_m) = b$$

Análogamente

$$Ax_2 = b + \alpha(\mathbf{0}_m) = b$$

Luego

$$\begin{aligned} x_1 &= x + \alpha\lambda \\ \underline{x_2 &= x - \alpha\lambda} \\ x_1 + x_2 &= 2x + \alpha\lambda - \alpha\lambda \Rightarrow x = \tfrac{1}{2}x_1 + \tfrac{1}{2}x_2 \end{aligned}$$

¡¡¡Contradicción!!! La igualdad anterior es falsa. Lo cierto es lo que queríamos demostrar. ¿Qué forma tiene el α ? Para esto vamos a considerar las formas de x_1 y x_2:

$$x_1 = x + \alpha\lambda$$

$$\begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1k} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \alpha \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_k \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Recordemos que $x_1 \geq 0$ por definición de poliedro, $x_{1j} = x_j + \alpha\lambda_j$ con $\alpha > 0$, $j = 1, 2, ..., k$

Si $\lambda_j \geq 0, \forall j = 1, 2, ..., k$, entonces $x_{1j} \geq 0$.
Si $\lambda < 0, \forall j = 1, 2, ..., k$, entonces $x_{1j} = x_j + \alpha\lambda_j \geq 0 \Rightarrow x_j \geq \alpha(-\lambda_j)$

$$\Rightarrow \tfrac{x_j}{-\lambda_j} \geq \alpha$$

Como $\lambda_j < 0 \Rightarrow -1 > \lambda_j > 0$, por definición de valor absoluto:

$$\lambda_j < 0 \Rightarrow |\lambda_j| > 0$$

luego

$$\tfrac{x_j}{|\lambda_j|} \geq \alpha \quad \text{o} \quad \alpha \leq \tfrac{x_j}{|\lambda_j|}$$

El análisis es el mismo para x_2. Por otro lado, como $j = 1, 2, ..., k$

$$\alpha = \min_{1 \leq j \leq k} \left\{ \frac{x_j}{|\lambda_j|}, \lambda_j \neq 0 \right\}$$

Luego, tenemos que $\{a_j\}_{j=1}^{k}$ son L.I. pero $rg(A) = m$, A es de rango completo, entonces k está obligado a ser $k \leq m$. Si $k = m$ entonces no hay problema. Si $k < m$ entonces podemos completar $\{a_1, a_2, ..., a_k\}$ con $(m - k)$ vectores L.I. a partir de las columnas de A, para obtener B. ■

Corolario 31 *El número de puntos extremos de un poliedro de la forma* $S = \{x \in \mathbb{R}^n / Ax = b, x \geq \mathbf{0}_n\}, A_{mxn}, b \in \mathbb{R}^m$*, es finito.*

Teorema 32 ***(Teorema de Existencia de Puntos Extremos)*** *Sea* $S = \{x \in \mathbb{R}^n / AX = b, x \geq \mathbf{0_n}\}, A_{mxn}, rg(A) = m$*,* $b \in \mathbb{R}^m \Rightarrow S$ *tiene al menos un punto extremo.*

Demostración
Si $S \neq \emptyset$, existirá al menos un $x \in S$. Sin pérdida de generalidad supongamos que:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ \vdots \\ x_n \end{bmatrix} \; con \; k \geq 1$$

Ya que si $k = 0$, entonces $x = \mathbf{0_n}$ y evidentemente x sería un punto extremo (por las condiciones de no negatividad el nulo espunto extremo y también por que no se puede expresar como combinación convexa estricta de dos puntos x_1 y x_2 no nulos).

Si $a_1, a_2, ..., a_k$ fuesen linealmente independiente por el teorema de caracterización de puntos extremos tendríamos que x es un punto extremo.
$A = [\underbrace{a_1, a_2, ..., a_k}_{columnas\ asociadas\ a\ x_1, x_2, ..., x_k}, a_{k+1}, ..., a_n]$
$x = [x_1, x_2, ..., x_k, x_{k+1}, ...x_n]$

Supongamos $\{a_j\}_{j=1}^{k}$ no son LI y veamos como acceder a un punto extremo desde x (El hecho que las a_j no sean LI lleva a que x no sea un punto extremo).

Si las $\{a_j\}_{j=1}^{k}$ son LD exsitirán $\lambda_1, \lambda_2, ..., \lambda_k \in \mathbb{R}$ no todos nulos tales que $\mathbf{0} = \sum_{j=1}^{k} \lambda_j a_j$ esta forma podemos suponer que al menos uno de ellos es positivo.
Sea

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R} \quad y \quad \alpha = \min_{1 \leq j \leq k} \quad \left\{ \frac{x_j}{\lambda_j} : \lambda_j > 0 \right\} = \frac{x_h}{\lambda_h} > 0 \Rightarrow \alpha = \frac{x_h}{\lambda_h} > 0$$

Si calculamos α de la forma anterior, nos garantizamos que al construir el punto $x' = x - \alpha\lambda \in S$ y que $x'_h = x_h - \alpha\lambda_h = 0$

Veamos

$$Ax' = A(x - \alpha\lambda) = Ax - \alpha A\lambda = Ax - \alpha[a_1, ..., a_n] \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix} = Ax - \alpha \sum_{j=1}^{k} a_j \lambda_j$$

$$Ax' = Ax - \alpha \sum_{j=1}^{k} \lambda_j a_j = Ax - \alpha(0) = Ax = b$$

También

$$x'_h = x_h - \alpha\lambda_h = x_h - \frac{x_h}{\cancel{\lambda_h}}\cancel{\lambda_h} = 0$$

La idea de hacer $x' = x - \alpha\lambda$, es la de proyectar en uno de los ejes y por tanto una de las componentes es cero.

Observemos cómo funciona esto:
$x' = x - \alpha\lambda$

$$\begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_h \\ \vdots \\ x'_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_h \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \alpha \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_h \\ \vdots \\ \lambda_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Pero $\alpha = \frac{x_h}{\lambda_h} \Rightarrow \alpha\lambda_h = \frac{x_h}{\cancel{\lambda_h}}\cancel{\lambda_h} = x_h$, luego $x_h - x_h = 0$. Luego x' tiene como mínimo una componente positiva menos que x. Si x' es un punto extremo, i.e., las columnas asociadas a sus componentes

positivas son LI, entonces habremos terminado. En otro caso repetimos el proceso partiendo ahora de x'. Es evidente que en un número de pasos llegaremos a un punto extremo.

Lo anterior significa lo siguiente: si al realizar el procedimiento anterior x' no es un punto extremo, significa que hay columnas asociadas a sus $k-1$ componentes positivas que son LD por tanto todavía no tenemos una base y habrá que seguir haciendo componentes ceros a partir de x'. ■

Lema 33 *Un vector* $d \in \mathbb{R}^n \backslash \{\mathbf{0}_n\}$ *es una dirección de* $S = \{x \in \mathbb{R}^n / AX = b, x \geq \mathbf{0_n}\}$, $A_{mxn}, rg(A) = m$,$b \in \mathbb{R}^m$ *si y solo si* $Ad = \mathbf{0_m}$ *y* $d \geq \mathbf{0}_n$.

Demostración $(\Rightarrow)$
Sabemos que d es dirección de $S \Rightarrow \forall x \in S, \forall \lambda \geq 0$, $x + \lambda d \in S$, también que si d es dirección de $S \Rightarrow A(x + \lambda d) = b$, luego $A(x + \lambda d) = Ax + \lambda Ad = b$, que se cumplirá si y solo si $\lambda Ad = \mathbf{0}_m$, pero $\lambda \geq 0$ esto lleva a que necesariamente $Ad = \mathbf{0}_m$ lo que se cumple tanto si $\lambda = 0$ o $\lambda > 0$.

Ahora nos preguntamos si $x + \lambda d \geq 0$. Por hipótesis $x \in S \Rightarrow x \geq 0$, también $\lambda \geq 0$. Por tanto nos interesa saber que condiciones debemos imponerle a d para que $(x + \lambda d) \geq 0$, lo único que sabemos es que $d \neq \mathbf{0}$. Si $d \geq \mathbf{0}_n$, entonces no habría nada que probar dado que $(x + \lambda d) \geq 0$. Supongamos que d tiene al menos una componente negativa. Sea esta d_j y que $(x_j + \lambda d_j) \geq 0$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{bmatrix} + \lambda \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_j \\ \vdots \\ d_n \end{bmatrix} \Rightarrow x_j + \lambda d_j \geq 0?$$

Qué debemos exigirle a d_j para que la expresión anterior sea mayor o igual a cero.
luego $x_j + \lambda d_j \geq 0 \Rightarrow \lambda d_j \geq -x_j \Rightarrow \lambda \geq \frac{-x_j}{d_j}$ con $d_j < 0 \Rightarrow \lambda = \frac{-x_j}{d_j} \geq 0 \Rightarrow \lambda \geq \frac{-x_j}{d_j}$, ya que $x_j \geq 0$

Si $\lambda = \frac{-x_j}{d_j}$, entonces $x_j + \lambda d_j = x_j + \frac{-x_j}{d_j} = x_j - x_j = 0$

Si $\lambda > \frac{-x_j}{d_j}$, entonces $x_j + \lambda d_j = x_j + \alpha[-\frac{x_j}{d_j}]d_j = x_j - \alpha[\frac{x_j}{d_j}]d_j = x_j - \alpha x_j = (1 - \alpha)x_j$
$x_j + \lambda d_j = x_j + (-\frac{x_j}{d_j})d_j = x_j + \alpha(-x_j) = x_j - \alpha x_j$

Como $\lambda \geq 0$, cuando $\lambda > 1 \Rightarrow x_j + \lambda d_j < 0$
$x_j + \lambda d_j = (1 - \alpha)x_j$, pero $1 - \alpha < 0$ para $\alpha > 1$, entonces $x_j + \lambda d_j < 0$ ¡¡¡Contradicción!!!

$\therefore d_j \geq 0, \forall j \Rightarrow d \geq \mathbf{0}_n$

Demostración ($\Leftarrow$)
$Ad = \mathbf{0}_m$ y $d \geq \mathbf{0}_n \Rightarrow d \neq \mathbf{0}_n$ dirección de S, $S = \{x \in \mathbb{R}^n / Ax = b, x \geq \mathbf{0}_n\}$, A_{mxn},$rg(A) = m$,$b \in \mathbb{R}^m$
Aquí debemos probar que d es dirección de S.
Como $x \in S \Rightarrow Ax = b$

$$Ad = \mathbf{0}_m \Rightarrow Ax + \mathbf{0}_m = b$$
$$Ax + Ad = b$$
$$Ax + \lambda Ad = b$$
$$A(x + \lambda d) = b$$

Pero $(Ad) = \mathbf{0}_m$, por hipótesis también $\lambda Ad = \mathbf{0}_m$

Con esto se comprueba que $x + \lambda d$ sastiface $Ax = b$

Ahora nos falta saber si $(x + \lambda d) \geq \mathbf{0}_n$, esto es evidente porque $x \geq \mathbf{0}_n$, $d \geq \mathbf{0}_n$, $\lambda \geq 0$, por tanto $x + \lambda d \geq \mathbf{0}_n$, luego sabiendo que $(x + \lambda d)$ sastiface $Ax = b$ y que $(x + \lambda d) \geq \mathbf{0}_n$

Concluimos que $(x + \lambda d) \in S, \forall x \in S$ y $\forall \lambda \geq 0$

Aplicamos ahora la definición de dirección y concluimos que d es dirección de S. ■

Teorema 34 ***Caracterización de Direcciones Extremas.***
Dado $S = \{x \in \mathbb{R}^n / AX = b, x \geq \mathbf{0_n}\}, A_{mxn}, rg(A) = m$,$b \in \mathbb{R}^m$ un poliedro en donde A es una matriz mxn de rango completo y $b \in \mathbb{R}^m$, $\bar{d}$ es una dirección extrema de S si y solo si $Ax = \left[B \vdots N\right]$ con $B^{-1}a_j \leq 0$, para algún $j, a_j \in N$, y $\bar{d}$ es un múltiplo positivo de la dirección d, donde

$$d = \begin{bmatrix} -B^{-1}a_j \\ e_j \end{bmatrix} \quad siendo \quad e_j = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix}, a_j \in N.$$

Demostración ($\Leftarrow$)
El lema anterior dice que de $\mathbb{R}\backslash\{\mathbf{0}_n\}$ dirección de S si y solo si $Ad = \mathbf{0}_m$ y $d \geq \mathbf{0}_n$

Aquí debemos probar que d es dirección extrema o sea ($\bar{d} \geq \mathbf{0}_n, a\bar{d} = \mathbf{0}_m$ y $\bar{d}$) extrema.

Por hipótesis $\bar{d} = \alpha d$ con $\alpha > 0$, d dirección del poliedro S como $\alpha > 0$ y $d \neq \mathbf{0}_n \Rightarrow \bar{d} = \alpha d \neq \mathbf{0}_n$

Como

$$d = \begin{bmatrix} -B^{-1}a_j \\ e_j \end{bmatrix},\ entonces\ \bar{d} = \alpha d = \alpha \begin{bmatrix} -B^{-1}a_j \\ e_j \end{bmatrix} = \begin{bmatrix} \alpha(-B^{-1}a_j) \\ \alpha e_j \end{bmatrix}$$

Con $-B^{-1}a_j > \mathbf{0}_m$, ya que $B^{-1}a_j \leq 0$ y $\alpha(B^{-1}a_j)$, ya que $\alpha > 0$

También $e_j \geq \mathbf{0}_{n-m} \Rightarrow \alpha e_j \geq \mathbf{0}_{n-m}$

Por tanto

$$\bar{d} = \alpha d = \begin{bmatrix} \alpha(-B^{-1}a_j) \\ \alpha e_j \geq \mathbf{0}_n \end{bmatrix}$$

Finalmente $\bar{d} \geq \mathbf{0}_n$

Ahora probamos que $A\bar{d} = \mathbf{0}_m$

$$A\bar{d} = A(\alpha d) = \alpha A d = \alpha [B, N] \begin{bmatrix} -B^{-1}a_j \\ e_j \end{bmatrix} = \alpha \begin{bmatrix} -BB^{-1}a_j + Ne_j \end{bmatrix}$$
$$A\bar{d} = \alpha \begin{bmatrix} -Ia_j + a_j \end{bmatrix} = \alpha \begin{bmatrix} -a_j + a_j \end{bmatrix} = \alpha \mathbf{0}_m = \mathbf{0}_m$$

Finalmente $A\bar{d} = \mathbf{0}_m$

Nota:

$$Ne_j = \begin{bmatrix} a_{m+1}, a_{m+2}, \ldots a_j, \ldots, a_n \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = a_{m+1}(0) + a_{m+2}(0) + \ldots + a_j(1) + \ldots + a_n(0) = a_j$$

Ahora solo nos falta probar que $\bar{d}$ es extrema:

Supongamos que $d = \lambda_1 d_1 + \lambda_2 d_2$ con $d_1 \neq d_2$, $d_1, d_2 \in S \Rightarrow \bar{d} = \alpha d = \alpha(\lambda_1 d_1 + \lambda_2 d_2)$,$\lambda_1, \lambda_2 > 0$

$$\bar{d} = \alpha \begin{bmatrix} -BB^{-1}a_j \\ e_j \end{bmatrix} = \alpha(\lambda_1 d_1 + \lambda_2 d_2)$$

$$\bar{d} = \alpha \begin{bmatrix} -BB^{-1}a_j \\ e_j \end{bmatrix} = \alpha\lambda_1 d_1 + \lambda_2 d_2 = \beta_1 d_1 + \beta_2 d_2$$

$$\bar{d} = \beta_1 d_1 + \beta_2 d_2$$

$$\underbrace{\begin{bmatrix} \left.\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}\right\} \text{m} \\ \vdots \\ \left.\begin{pmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}\right\} \text{n-m} \end{bmatrix}}_{} = \beta_1 \underbrace{\begin{bmatrix} \left.\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}\right\} \text{m} \\ \vdots \\ \left.\begin{pmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}\right\} \text{n-m} \end{bmatrix}}_{} + \beta_2 \underbrace{\begin{bmatrix} \left.\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}\right\} \text{m} \\ \vdots \\ \left.\begin{pmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}\right\} \text{n-m} \end{bmatrix}}_{}$$

Este vector d tiene $n - m - 1$ componentes iguales a cero

Las $n - m - 1$ componentes de d_1 y d_2 asociadas a las $n - m - 1$ componentes de d son necesariamente ceros, por lo tanto podemos hacer lo siguiente:

$$\bar{d} = \alpha \begin{bmatrix} -B^{-1}a_j \\ e_j \end{bmatrix} = \alpha\left(\lambda_1 \begin{bmatrix} d_1' \\ e_1 e' \end{bmatrix} + \lambda_2 \begin{bmatrix} d_2' \\ e_2 e'' \end{bmatrix}\right)$$

Como d_1 y d_2 son direcciones de S, entonces $Ad_1 = \mathbf{0}_m$

$$[B, N] \begin{bmatrix} d_1' \\ e_j \end{bmatrix} = Bd_1' + Ne_j = Bd_1' + a_j = \mathbf{0}_m \Rightarrow Bd_1' = \mathbf{0}_m - a_j = -a_j$$
$$Bd_1' = -a_j \Rightarrow d_1' = -B^{-1}a_j$$

de la misma forma $d_2' = -B^{-1}a_j$
De lo anterior

$$d_1' = d_2' \Rightarrow d_1 = d_2 = d = \begin{bmatrix} -B^{-1}a_j \\ e_j \end{bmatrix}$$

y por tanto

$$\bar{d} = \alpha \begin{bmatrix} -B^{-1}a_j \\ e_j \end{bmatrix} = \alpha(\lambda_1 d_1 + \lambda_2 d_1) = \alpha d_1(\lambda_1 + \lambda_2) = \alpha d_1(1) = \alpha d_1$$

luego

$$\bar{d} = \alpha \begin{bmatrix} -B^{-1}a_j \\ e_j \end{bmatrix} = \alpha d$$

$\therefore \bar{d} = \alpha d$ es dirección extrema.

Demostración $(\Rightarrow)$.
Por hipótesis $\bar{d}$ es una dirección extrema de S, entonces $\bar{d} \neq \mathbf{0}_n$. Y por el lema anterior $\bar{d} \geq \mathbf{0}_n$ y $A\bar{d} = \mathbf{0}_m$.

Dado que $\bar{d} \geq \mathbf{0}_n$ existirá al menos una componente en $\bar{d}$ positiva vamos a suponer que la j-ésima $\bar{d}_j$

$$\bar{d} = \begin{bmatrix} \bar{d}_1 \\ \bar{d}_2 \\ \vdots \\ \bar{d}_k \\ 0 \\ \vdots \\ 0 \\ \bar{d}_j \\ 0 \\ \vdots \\ 0 \end{bmatrix} con \bar{d}_1, \bar{d}_2, ..., \bar{d}_k, \bar{d}_j > 0$$

Como al menos una componente en $\bar{d}$ debe ser positiva, sean estas $\bar{d}_1, \bar{d}_2, ..., \bar{d}_k, \bar{d}_j$, donde hemos dejado las k primeras componentes positivas ubicadas de primeras.

Por RAA supongamos que no lo son, entonces existirán $\lambda_1, \lambda_2, ..., \lambda_k$ escalares no nulos tales que

$$\sum_{j=1}^{k} \lambda_j a_j = \mathbf{0}_m$$

Utilizaremos la misma estrategia que se usó en la demostración del teorema de existencia de puntos extremos.

$$\bar{d} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Como los λ no son todos nulos, ordenamos los $\lambda > 0$ en las primeras componentes y entonces λ será una dirección de $\mathbb{R}^n$

Por construcción como $\bar{d} \in \mathbb{R}^n$ y $\lambda \in \mathbb{R}^n$ es una dirección de $\mathbb{R}^n$, entonces podemos hacer lo siguiente:

$$\begin{cases} d_1 = \bar{d} - \alpha_1 \lambda \\ d_2 = \bar{d} + \alpha_2 \lambda \end{cases}$$

Ahora analizamos que valor puede tomar α para que d_1 y d_2 sea direcciones de S. Para que d_1 y d_2 sea direcciones de S, deben cumplir $d_1 \geq \mathbf{0}_n$, $d_2 \geq \mathbf{0}_n$ y $Ad_1 = \mathbf{0}_m$, $Ad_2 = \mathbf{0}_m$
Amplifiquemos con d_1

$$d_1 = \bar{d} - \alpha_1 \lambda$$

$$\underbrace{\begin{bmatrix} d_{11} \\ d_{12} \\ \vdots \\ d_{1k} \\ d_{1(k+1)} \\ \vdots \\ d_{1n} \end{bmatrix}}_{d_1} = \underbrace{\begin{bmatrix} \bar{d}_1 \\ \bar{d}_2 \\ \vdots \\ \bar{d}_k \\ \bar{d}_{(k+1)} \\ \vdots \\ \bar{d}_n \end{bmatrix}}_{\bar{d}} - \alpha_1 \underbrace{\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_K \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\alpha_1 \lambda} \Bigg\} \text{r componentes}$$

Si asumimos que d_1 es dirección, entonces a partir de la componente $(k+1)$ tendremos

$$
\begin{aligned}
&d_{1r} = \bar{d}_r - \alpha\lambda r, \quad r = (k+1), (k+2), ..., n \\
&d_{1r} = \bar{d}r - \alpha(0) \\
&d_{2r} = dr \neq 0 \\
&d_{2r} = \bar{d}r - \alpha\lambda r \\
&d_{1r} = dr - \alpha(0) = dr \neq 0
\end{aligned}
$$

Todo lo anterior fue para $(k+1) \leq r \leq n$.

Ahora el análisis se hace para $1 \leq r \leq k$ sabiendo que los λ son escalares no todos nulos, o sea que hay positivos y negativos. Nos interesa que $d_1 \geq 0$ y $d_2 \geq 0$, entonces vamos a buscar los valores de α_1 α_2

$$d_1 = \bar{d} - \alpha_1\lambda$$

$$
\left.\begin{bmatrix} d_{11} \\ \vdots \\ d_{1k} \\ 0 \\ \vdots \\ 0 \\ d_j \\ \vdots \\ 0 \end{bmatrix}\right\} = \left.\begin{bmatrix} \bar{d}_1 \\ \vdots \\ \bar{d}_k \\ 0 \\ \vdots \\ 0 \\ \bar{d}_j \\ \vdots \\ 0 \end{bmatrix}\right\} - \alpha_1 \left.\begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_k \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right\} 1 \leq r \leq k
$$

Para $\lambda_r < 0$,$1 \leq r \leq k \Rightarrow d_{1r} = \bar{d}_r - \alpha_1\lambda r > 0$
Para $\lambda_r = 0$,$1 \leq r \leq k \Rightarrow d_{1r} = \bar{d}_r - \alpha_1(0) = 0$
Para $\lambda_r > 0$,$1 \leq r \leq k \Rightarrow \alpha_1 = min\{\frac{\bar{d}_r}{\lambda r}, \lambda_r > 0\}$
Para $d_2 = \bar{d} + \alpha_2\lambda$, hacemos lo siguiente
Para $\lambda_r \geq 0$, entonces $d_{2r} = \bar{d}_r + \alpha\lambda_r \geq 0$
Para $\lambda_r < 0$, entonces $d_{2r} = \bar{d}_r + \alpha\lambda_r \Rightarrow$ tomamos $\alpha_2 = min\{\frac{\bar{d}_r}{-\lambda r}, \lambda_r < 0\}$

Luego hacemos $\alpha = min\{\alpha_1, \alpha_2\}$ y con esto tenemos $d_1 \geq \mathbf{0}_n$, $d_2 \geq \mathbf{0}_n$.

Ahora verificamos que

$$Ad_1 = \mathbf{0}$$
$$Ad_1 = A(\bar{d} - \alpha\lambda) = \cancel{A\bar{d}} - \alpha A\lambda = -\alpha\lambda A = -\alpha[\sum_{r=1}^{n} a_r\lambda_r]$$
$$Ad_1 = -\alpha[\sum_{r=1}^{k} a_r\lambda_r + \sum_{r=k+1}^{n} a_r\lambda_r] = -\alpha[\mathbf{0}_m + \mathbf{0}_m] = \mathbf{0}_m$$

Por tanto hemos probado que

$$d_1 \geq \mathbf{0}_n, Ad_1 \geq \mathbf{0}_m$$
$$d_2 \geq \mathbf{0}_n, Ad_2 \geq \mathbf{0}_m$$

Luego d_1 y d_2 son direcciones de S

Ahora preguntaremos que si d_1 y d_2 son iguales

Supongamos que $d_2 = \beta d_1$, $\beta > 0$ d_2 es múltiplo positivo de d_1

$$d_{2j} = \beta d_{1j}, j = 1, 2, ..., n(*)$$

Con

$$d_{2j} = \bar{d}_j + \alpha\lambda_j, d_{1j} = \bar{d}_j - \alpha\lambda_j$$

Sustituyendo en (*)
Si $j < k$ y $\lambda_j \neq 0$

$$\bar{d}_j + \alpha\lambda_j = \beta(\bar{d}_j - \alpha\lambda_j)$$

Como $\beta > 0$, hagamos $\beta = 1 > 0$

$$\bar{d}_j + \alpha\lambda_j = (1)(\bar{d}_j - \alpha\lambda_j)$$
$$\cancel{\bar{d}_j} + \alpha\lambda_j = \cancel{\bar{d}_j} - \alpha\lambda_j$$
$$\alpha\lambda_j = -\alpha\lambda_j$$
$$\lambda_j = -\lambda_j$$
$$1 = -1 \quad Contradicción!!!$$

Luego d_2 no es un múltiplo positivo de d_1 y como $d_1 \neq d_2$ hacemos

$$d_1 = \bar{d} - \alpha\lambda$$
$$\underline{d_2 = \bar{d} + \alpha\lambda}$$
$$d_1 + d_2 = \bar{d} - \cancel{\alpha\lambda} + \bar{d} + \cancel{\alpha\lambda} = 2\bar{d} \Rightarrow \bar{d} = \tfrac{1}{2}d_1 + \tfrac{1}{2}d_2$$

¡¡¡Contradicción!!! con la hipótesis que dice que d es dirección extrema y no se puede expresar como combinación de dos direcciones distintas de S.

Luego lo asumido es falso, lo cierto es que $\{a_j\}_{j=1}^{k}$ son LI, como $rg(A) = m$, entonces $k \leq m$, si $k < m$ (menor estricto que m) podríamos completar con $(m-k)$ columnas de A distintas de a_j hasta tener m columnas linealmente independientes, pero diferentes de a_j.

¿Qué pasa si tomamos a_j? Supongamos que para completar los $\{a_j\}_{j=1}^{k}$ columnas LI, incluimos a_j para completar B

$$\bar{d} = \begin{bmatrix} \bar{d}_1 \\ \bar{d}_2 \\ \vdots \\ \bar{d}_k \\ \bar{d}_j \\ \bar{d}_m \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \bar{d}_B \\ \bar{d}_N \end{bmatrix} \quad Con \quad \bar{d}_N = \mathbf{0}_{n-m}$$

Pero $A\bar{d} = \mathbf{0}_m$. Esto por el lema anterior $Ad = \mathbf{0}_m$ y $d \geq \mathbf{0}_n$

$$[B, N] \begin{bmatrix} \bar{d}_B \\ \bar{d}_N \end{bmatrix} = \mathbf{0}_m$$

$$B\bar{d}_B + N\bar{d}_N = \mathbf{0}_m, \quad Con \quad N\bar{d}_N = [a_{m+1}, a_{m+2}, ..., a_n] \begin{bmatrix} \bar{d}_{m+1} \\ \vdots \\ \bar{d}_m \end{bmatrix} = \sum_{j=m+1}^{n} \underbrace{a_{m+1}\bar{d}_{m+1}}_{\mathbf{0}_m} = \mathbf{0}_m$$

$$\begin{aligned} & B\bar{d}_B + \mathbf{0}_m = \mathbf{0}_m \\ & B^{-1}B\bar{d}_B = B^{-1}\mathbf{0}_m \\ & I\bar{d}_B = \mathbf{0}_m \\ & \bar{d}_B = \mathbf{0}_m \Rightarrow \bar{d} = \begin{bmatrix} \bar{d}_B \\ \bar{d}_N \end{bmatrix} = \begin{bmatrix} \mathbf{0}_m \\ \mathbf{0}_{n-m} \end{bmatrix} = \mathbf{0}_m \end{aligned}$$

¡¡¡Contradicción!!! Por tanto no podemos completar B con a_j

De lo anterior tenemos que $a_j \notin B \Rightarrow a_j \in N$, luego $A\bar{d} = \mathbf{0}_m$

$$[B, N] \begin{bmatrix} \bar{d}_B \\ \bar{d}_N \end{bmatrix} = \mathbf{0}_m$$

$$B\bar{d}_B + N\bar{d}_N = \mathbf{0}_m$$

Ahora

$$N\bar{d}_N = [a_{m+1}, a_{m+2}, ..., a_j, ..., a_n] \begin{bmatrix} 0 \\ \vdots \\ \bar{d}_j \\ \vdots \\ 0 \end{bmatrix} = a_{m+1}(0) + \ldots + a_j\bar{d}_j + 0 + \ldots + 0 = a_j\bar{d}_j$$

$$\begin{aligned} &B\bar{d}_B + a_j\bar{d}_j = \mathbf{0}_m \\ &B\bar{d}_B = -a_j\bar{d}_j \\ &\bar{d}_B = -B^{-1}a_j\bar{d}_j \end{aligned}$$

Como $\bar{d}_B \geq 0 \Rightarrow -B^{-1}a_j d_j \geq \mathbf{0}_m$
Como $d_j > 0 \Rightarrow -B^{-1}a_j \geq \mathbf{0} \Rightarrow -B^{-1}a_j < 0$
luego

$$\bar{d} = \begin{bmatrix} \bar{d}_B \\ \cdots \\ 0 \\ \vdots \\ \bar{d}_j \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} -B^{-1}a_j\bar{d}_j \\ \cdots \\ 0 \\ \vdots \\ \bar{d}_j \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} -B^{-1}a_j \\ \cdots \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \bar{d}_j \begin{bmatrix} -B^{-1}a_j \\ \cdots \\ e_j \end{bmatrix}$$

$\bar{d}_j = \bar{d}_j \begin{bmatrix} -B^{-1}a_j \\ e_j \end{bmatrix}$, es decir $\bar{d}$ es múltiplo positivo de $\begin{bmatrix} -B^{-1}a_j \\ e_j \end{bmatrix}$ ■

Teorema 35 ***Representación Interna***
Sea $S = \{x \in \mathbb{R}^n / AX = b, x \geq \mathbf{0_n}\}, A_{mxn}, rg(A) = m$, $b \in \mathbb{R}^m$. *Sea* $\{v_1, v_2, ..., v_{|I|}\} = extr(S)$, $\{d_1, d_2, ..., d_{|J|}\} = Dextr(S)$, *entonces cualquier punto* $x \in S$ *puede expresarse de la forma*

$$\begin{aligned} &x = \sum_{i \in I} \lambda_i v_i + \sum_{j \in J} \mu_j d_j \\ &\lambda_i \geq 0, \forall i, \sum_{i \in I} \lambda_i = 1 \\ &\mu_j \geq 0, \forall j \in J \end{aligned}$$

Demostración
Haremos la demostración por inducción sobre el número de componentes positivas, k de $x^t = [x_1, x_2, ..., x_k, 0, ..., 0]$

Si $k = 0$, entonces $x = \mathbf{0}_n$ [Si no hay ningún componente positiva, entonces todas serán cero], luego $b = \mathbf{0}_m$, ya que $Ax = b$. De lo anterior tenemos que: $x = 1 \cdot v_1$, $\lambda_i = 0, \forall i > 1$, $\mu_j = 0, \forall j$

Supongamos que el resultado es cierto para todo x con menos de k componentes positivas y veamos que es cierto para un x con k componentes positivas.

$$x^t = [x_1, x_2, ..., x_k, 0, ..., 0]$$

Si $x = v_i$ para algún i, entonces el resultado es trivial, en otro caso x no es punto extremo. Por el teorema de caracterización de puntos extremos, las columnas de A asociadas a las componentes positivas de x son LD, luego $\exists w \in \mathbb{R}^n$, $w \neq \mathbf{0}_n$, $Aw = \mathbf{0}_m$, $w_i = 0$, Si $x_i = 0$.

$$\sum_{r=1}^{k} w_i a_i = 0$$

Se distinguen tres situaciones dependiendo si w tiene o no componentes positivos y/o negativas.

a) w tiene componentes de ambos signos, hacemos $x'' \dashrightarrow x \dashrightarrow x'$

$$x'' = x - \beta w, \beta > 0 \quad x' = x + \alpha w, \alpha > 0$$

Debemos hallar los valores de α y β para que x' y $x'' \in S$ y tengan a lo sumo $(k-1)$ componentes no nulas.

$$Ax' = A(x + \alpha w) = Ax + \alpha Aw = b + \alpha \mathbf{0} = b$$

También $Ax'' = b$

Ahora veamos que $\forall i, w_i \geq 0 \Rightarrow x'_i \geq 0$

Si $w_i < 0$, entonces $x'_i \geq 0 \Leftrightarrow x' = x_i + \alpha w_i \geq 0$, $\alpha \leq \frac{x_i}{-w_i}$

$$0 < \alpha = min\{\tfrac{x_i}{-w_i} : w_i < 0\} = \tfrac{x_h}{-w_h}$$

Luego

$$x'_h = x_h + \alpha w_h = x_h - \frac{x_h}{\cancel{w_h}}\cancel{w_h} = x_h - x_h = 0$$

Con lo anterior hacemos una comnponente igual a cero y nos garantizamos que x' tendrá a lo sumo $(k-1)$ componentes positivas.

Ahora probamos para x''

$$\begin{aligned}
&\forall i, w_i \leq 0 \Rightarrow x''_i = x_i - \beta w_i \geq 0 \\
&Si \quad w_i > 0, \quad entonces \quad x''_i \geq 0 \Leftrightarrow x_i - \beta w_i \geq 0 \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad -\beta \geq \tfrac{-x_i}{w_i} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \beta \leq \tfrac{x_i}{w_i} \\
&0 < \beta = min\{\tfrac{x_i}{w_i} : w_i > 0\} = \tfrac{x_r}{w_r} \\
&x'' = x_r - \beta w_r = x_r - \frac{x_r}{\cancel{w_r}}\cancel{w_r} = x_r - x_r = 0
\end{aligned}$$

Con lo anterior hacemos una componente igual a cero y nos garantizamos que x'' a lo sumo $(k-1)$ componentes positivas.

Resumiendo lo anterior tenemos:

$x' = x + \alpha w, \alpha > 0$	$x'' = x - \beta w, \beta > 0$
$Ax' = b$	$Ax'' = b$
x' tiene a lo sumo $(k-1)$ componentes positivas.	x'' tiene a lo sumo $(k-1)$ componentes positivas.

Por hipótesis de inducción ocurre que:

$$x' = \sum_{i \in I} \lambda'_i v_i + \sum_{j \in J} \mu'_j d_j, \quad \mu'_j \geq 0, \quad \forall j \in J, \quad \lambda'_i \geq 0, \quad \forall i \in I, \quad \sum_{i \in I} \lambda'_i = 1$$
$$x'' = \sum_{i \in I} \lambda''_i v_i + \sum_{j \in J} \mu''_j d_j, \quad \mu''_j \geq 0, \quad \forall j \in J, \quad \lambda''_i \geq 0, \quad \forall i \in I, \quad \sum_{i \in I} \lambda''_i = 1$$

x es cualquier punto del poliedro y como x' y $x'' \in S$, entonces podemos hacer

$$x = ax' + bx'' \quad \text{con} \quad a + b = 1$$

Sea $a = \frac{\beta}{\alpha+\beta}$ y $b = \frac{\alpha}{\alpha+\beta}$

$$\begin{aligned}
&x = \tfrac{\beta}{\alpha+\beta}x' + \tfrac{\alpha}{\alpha+\beta}x'' \quad es\ evidente\ que \quad \tfrac{\alpha}{\alpha+\beta} + \tfrac{\beta}{\alpha+\beta} = \tfrac{\alpha+\beta}{\alpha+\beta} = 1 \\
&x = \tfrac{\beta}{\alpha+\beta}(x + \alpha w) + \tfrac{\alpha}{\alpha+\beta}(x - \beta w)
\end{aligned}$$

Sea $\mu = \frac{\alpha}{\alpha+\beta}, \quad 1-\mu = \frac{\beta}{\alpha+\beta}$

$$x = (1-\mu)x' + \mu x'' \quad con \mu = \frac{\alpha}{\alpha+\beta} \in (0,1)$$
$$x = (1-\mu)\sum_{i\in I}\lambda_i' v_i + \sum_{j\in J}(1-\mu)\mu_j' d_j + \sum_{i\in I}\mu\lambda_i'' v_i + \sum_{j\in J}\mu\mu_j'' d_j$$
$$x = \sum_{i\in I}\underbrace{[(1-\mu)\lambda_i' + \mu\lambda'']v_i}_{\geq 0} + \sum_{j\in J}\underbrace{[(1-\mu)\mu_j' + \mu\mu_j'']}_{\geq 0} d_j$$

Donde

$$(1-\mu)\lambda_i' + \mu\lambda'' = (1-\mu)\sum_{i\in I}\lambda_i + \mu\sum_{i\in I}\lambda_i'' = (1-\mu)(1) + \mu(1) = 1-\mu+\mu = 1$$

Luego hemos expresado x como combinación convexa de los puntos extremos más una combinación no negativa de las direcciones extremas del poliedro.

b) Las componentes son $w \leq \mathbf{0}_n$, $w \neq \mathbf{0}_n$. Como $Aw = \mathbf{0}_n$, entonces $(-w)$ es dirección de S y tenemos que $x' = x + \alpha w \Rightarrow x = x' + \alpha(-w)$

c) Si $w \geq \mathbf{0}_n$, $w \neq \mathbf{0}_n$. w es dirección de S y tenemos $x'' = x - \beta w \Rightarrow x = x''' + \beta(w)$
Por inducción sobre las componentes positivas de x, x cualquier punto de S

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha d$$

$k = 0 \Rightarrow x = \mathbf{0}_n \Rightarrow x = 1v_1 \quad$ con $\quad \lambda_i = 0, \forall i > 1$

Supongamos cierto para un número de componentes positivas menor que k

$$x = \left.\begin{bmatrix} x_1 \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right\} \text{menos de } k \text{ componentes positivas}$$

Si $x = v_i$ para algún i, el resultado es trivial. En caso contrario x no es punto extremo y las componentes de A asociadas a las componentes positivas de x son LD, entonces $\exists w' \in \mathbb{R}^n$ talque $Aw = \mathbf{0}$, $w \neq \mathbf{0}_n$

Se analizan los casos w tiene componentes positivas y/o negativas $w_i \leq \mathbf{0}_n$, $w \neq \mathbf{0}_n$
$w_i \geq \mathbf{0}_n$,$w \neq \mathbf{0}_n$. ■

9.1.3 Fundamentos Matemáticos del Simplex

9.1.3.1 Algoritmo del Método Simplex.

$$\begin{array}{ll} \text{Minimizar} & cx \\ \text{s.a} & Ax = b, \quad A_{mxn}, \quad rg(A) = m \\ & x \geq 0 \end{array}$$

Problema de minimización
Paso principal

1. Resuelve el sistema $BX_B = b \Rightarrow X_B = B^{-1}b = \bar{b}$
   Sea $X_B = \bar{b}$, $X_N = 0$, $Z = C_B X_B$

2. Para todas las variables no básicas calcúlese $Z_j - C_j = C_B B^{-1} a_j - C_j$ con $Z_k - a_k = max_{j \in N}(Z_j - C_j)$. En donde N es el conjunto de índices asociados con las variables no básicas. Si $Z_k - C_k \leq 0$, entonces el proceso se detiene con la solución básica factible presente como una solución óptima. En caso contrario, se sigue al paso 3)

3. Resuelva el sistema $y_k = B^{-1}a_k$. Si $y_k \leq 0$, entonces el proceso se detiene con la conclusión que la solución óptima es no acotada a lo largo del rayo

$$\left\{ \begin{bmatrix} \bar{b} \\ 0 \end{bmatrix} + \begin{bmatrix} -y_k \\ e_k \end{bmatrix} x_k : x_k \geq 0 \right\}$$

   En donde e_k es un $n - m$ vector de ceros excepto por un 1 en la k-ésima posición. Si $y_k \geq 0$, se sigue al paso 4).

4. Ahora, x_k entra a la base y la variable de bloqueo x_{B_r} sale de ella en donde el índice r se determina mediante el siguiente criterio de la razón mínima.

$$\frac{\bar{b}_r}{y_{rk}} = min_{1 \leq i \leq m} \left\{ \frac{\bar{b_i}}{y_{ik}} : y_{ik} > 0 \right\}$$

Luego se modifica la base B en donde a_k reemplaza a a_{Br}, se actualiza el conjunto de índices y se repite el paso 1)

9.1.3.2 Convergencia Finita del Método Simplex en la Ausencia de Degeneración

En cada iteración del método simplex se efectúa una de las tres acciones siguientes:

1. El proceso se detiene con un punto extremo óptimo si $z_k - c_k \leq 0$

2. El proceso se detiene con una solución no acotada si $z_k - c_k > 0 \quad \text{y} \quad y_k \leq 0$

3. Se genera una nueva solución básica factible si $z_k - c_k > 0 \quad \text{y} \quad y_k \geq 0$

En ausencia de degeneración, $b_r \bar{>} 0$, y en consecuencia $x_k = \frac{\bar{b_r}}{y_{rk}} > 0$

Por tanto, la diferencia entre los valores objetivos de la iteración anterior y la presente es $(z_k - c_k)x_k$, ya que z disminuye de la siguiente forma:

$z = z_0 - (z_k - c_k)x_k$, en otras palabras, la función objetivo decrece estrictamente en cada iteración y en consecuencia, las soluciones básicas factibles generadas por el método simplex son distintas.

Puesto que solo hay un número finito de soluciones básicas factible, el método se detendrá en un número finito de pasos, ya sea con una solución óptima finita, o bien, con una solución óptima no acotada.

9.1.3.3 El Método Simplex en Formato Tabla

$$\begin{array}{ll} \text{Minimizar} & cx \\ \text{s.a} & Ax = b \\ & x \geq 0 \end{array}$$

Si x es una solución posible básica con base B, entonces el PL se puede representar de la siguiente forma:

$$\begin{array}{ll} \text{Minimizar} & cx \\ \text{s.a} & z - cx = 0 \\ & z - [c_B, c_N] \begin{bmatrix} x_B \\ x_N \end{bmatrix} = 0 \\ & z - c_B x_B - c_N x_N = 0 \\ & Ax = b \\ & [B, N] \begin{bmatrix} x_B \\ x_N \end{bmatrix} = b \\ & Bx_B + Nx_N = b \\ & x_B \geq 0, \quad x_N \geq 0 \end{array}$$

Resumiendo

$$\begin{array}{lll} \text{Minimizar} & z & \\ \text{s.a} & z - c_B x_B - c_N x_N = 0 & (1) \\ & Bx_B - Nx_N = b & (2) \\ & x_B \geq 0, \quad x_N \geq 0 & \end{array}$$

De (2) tenemos que:

$$\begin{array}{ll} x_B + B^{-1}Nx_N = B^{-1}b & (2') \\ c_B(x_B + B^{-1}Nx_N) = c_B B^{-1}b & \\ c_B x_B + c_B B^{-1}Nx_N = c_B B^{-1}b & (3) \end{array}$$

Sumando (1) con (3)

$$\begin{array}{r} z - c_B x_B - c_N x_N = 0 \\ c_B x_B + c_B B^{-1}Nx_N = c_B B^{-1}b \\ \hline z - \cancel{c_B x_B} - c_N x_N + \cancel{c_B x_B} + c_B B^{-1}Nx_N = c_B B^{-1}b \end{array}$$

$$\begin{array}{ll} z + c_B B^{-1}Nx_N - c_N x_N = c_B B^{-1}b & \\ z + (c_B B^{-1}N - c_N)x_N = c_B B^{-1}b & \\ z + \mathbf{0}x_B + (c_B B^{-1}N - c_N)x_N = c_B B^{-1}b & (4) \end{array}$$

Con (2) y (4) tenemos

$$\begin{cases} z + \mathbf{0}x_B + (c_B B^{-1}N - c_N)x_N = c_B B^{-1}b \\ x_B + (B^{-1}N)x_N = B^{-1}b \end{cases} \tag{1}$$

Si $x_N = \mathbf{0}$ es evidente que

$$z = c_B B^{-1}b = c_B x_B$$
$$x_B = B^{-1}b$$

También se puede representar convenientemente la solución básica factible actual en la siguiente tabla. Aquí se piensa en z como una variable (básica) que debe minimizarse. La fila objetivo será la fila O y las filas restantes van de 1 a m la columna del lado derecho (Right Hand Side, RHS) denotará los valores de las variables básicas (incluyendo la función objetivo). Las variables básicas se identificarán en la columna del extremo izquierda. Notemos que:

	z	$\boldsymbol{x_B}$	$\boldsymbol{x_N}$	**RHS**
z	1	0	$c_B B^{-1}N - c_N$	$c_B B^{-1}b$
$\boldsymbol{x_B}$	0	I	$B^{-1}N$	$B^{-1}b$

Tabla 1: *Esta tabla da toda la información necesaria para desarrollar el método simplex.*

$$c_B B^{-1}N - c_N = c_B B^{-1}[a_{m+1}, a_{m+2}, \ldots, a_n] - [c_{m+1}, c_{m+2}, \ldots, c_{m+n}]$$
$$c_B B^{-1}N - c_N = [c_B B^{-1}a_{m+1}, c_B B^{-1}a_{m+2}, \ldots, c_B B^{-1}a_n] - [c_{m+1}, c_{m+2}, \ldots, c_{m+n}]$$
$$c_B B^{-1}N - c_N = [c_B B^{-1}a_{m+1} - c_{m+1}, c_B B^{-1}a_{m+2} - c_{m+2}, \ldots, c_B B^{-1}a_n - c_n]$$
$$c_B B^{-1}N - c_N = c_B B^{-1}a_j - c_j = z_j - c_j, \quad j \in N$$

También

$$B^{-1}N = B^{-1}[a_{m+1}, a_{m+2}, \ldots, a_n]$$
$$B^{-1}N = [B^{-1}a_{m+1}, B^{-1}a_{m+2}, \ldots, B^{-1}a_n]$$
$$B^{-1}N = [y_{m+1}, y_{m+2}, \ldots, y_n]$$
$$B^{-1}N = y_j, \quad j \in N$$

Por lo tanto podemos observar que:

1. En la fila O sabemos si hemos llegado al óptimo ya que consiste de los valores $z_j - c_j$ para las variables no básicas. Si los $z_j - c_j \geq 0$, $\forall j \in N$, tomaremos el máximo para determinar que variable entra a la base.

2. Si entra x_k (o sea se incrementa x_k), entonces el vector $y_k = B^{-1}a_k$ está representado en las filas $1, \ldots, m$ de la tabla bajo la variable x_k. Si $y_k \leq 0$, entonces x_k puede crecer indefinidamente sin ser bloqueada, y el objetivo óptimo es no acotado. Por otra parte si $y_k \geq 0$, es decir, si y_k

tiene al menos una componente positiva, entonces el incremento en x_k será bloqueado por una de las variables básicas actuales, las cual se hará cero. Como $y_k = B^{-1}a_k$ y $\bar{b} = B^{-1}b$ están disponibles en la tabla, es fácil determinar el criterio de la razón mínima.

El algoritmo hace lo siguiente:

1. Actualizar las variables básicas y sus valores

2. Actualizar los valores $z_j - c_j$ de las nuevas variables no básicas
   Actualizar las columnas y_j

Pivoteo. Todo lo anterior se puede realizar con una simple operación de pivoteo

1. Se divide la fila r por y_{rk}

2. Para $i = 1, ..., m$, con $i \neq r$, se actualiza la i-ésima fila multiplicada por $-y_{ik}$

3. Se actualiza la fila cero sumándola a la nueva r-ésima fila multiplicada por $c_k - z_k$

	z	x_{B1}	x_{B2}	$\cdots$	x_{Br}	$\cdots$	x_{Bm}	x_{m+1}	x_j	x_k	LD	
z	1	0	0	...	0	...	0	$z_{m+1} - c_{m+1}$	$z_j - c_j$	$z_k - c_k$	$c_B B^{-1} b$	
x_{B1}	0	1	0	...	0	...	0	...	y_{1j}	y_{1k}	$\bar{b1}$	
x_{B2}	.	.	1	...	0	...	0	...	.	.		
.	.	.	.	...	.	...	.	...	.	.		
.	.	.	.	...	.	...	.	...	.	.		
.	.	.	.	...	.	...	.	...	.	.		
x_{Br}	0	0	0	...	1	...	0	...	y_{rj}	$\boldsymbol{y_{rk}}$	$\bar{br}$	$\div y_{rk}$
.	.	.	.	...	0	...	.	...	.	.		
.	.	.	.	...	.	...	.	...	.	.		
.	.	.	.	...	.	...	.	...	.	.		
x_{Bm}	0	0	0	...	0	...	1	...	y_{mj}	y_{mk}	$\bar{bm}$	

Tabla 2: *Antes de pivotear*

1. Para hacer la posición y_{1k} cero $\quad 1(-y_{1k}) + y_{1k} = 0$

2. $\frac{y_{rj}}{y_{rk}}(-y_{1k}) + y_{1j} = y_{1j} - \frac{y_{rj}}{y_{rk}}y_{1k}$

3. Para hacer cero la posición $z_k - c_k \quad 1[-(z_k - c_k)] + (z_k - c_k) = 0$

4. $\frac{y_{rj}}{y_{rk}}[-(z_k - c_k)] + (z_j - c_j) = (z_j - c_j) - \frac{y_{rj}}{y_{rk}}(z_k - c_k)$

5. $\frac{y_1}{y_{rk}}[-(z_k - c_k)] + 0 = \frac{(c_k - z_k)}{y_{rk}}$

6. Para hacer cero cero la posición $y_{mj} \quad 1(-y_{mk}) + y_{mk} = 0$

7. $\frac{y_{rj}}{y_{rk}}(-y_{mk}) + y_{mj} = y_{mj - \frac{y_{rj}}{y_{rk}}(y_{mk})}$

8. $\frac{1}{y_{rk}}(-y_{mk}) + 0 = -\frac{y_{mk}}{y_{rk}}$

	z	x_{B1}		x_{Br}		x_{Bm}		x_j		x_k		RHS
z	1	.	.	$\frac{c_k - z_k}{y_{rk}}$	...	0	...	$(z_j - c_j)$ $-\frac{y_{rj}}{y_{rk}}(z_k - c_k)$	.	0	.	$c_B\bar{b} - (z_k - c_k)\frac{\bar{b_r}}{y_{rk}}$
	0	1	.	$-\frac{y_{1k}}{y_{rk}}$	...	0	...	$y_{1j} - \frac{y_{rj}}{y_{rk}}y_{1k}$	.	0	.	$\bar{b_1} - \frac{y_{1k}}{y_{rk}}\bar{b_r}$
	.	.	.	...	...	.	...	...	.		.	.
	.	.	.	...	...	.	...	...	.		.	.
	.	.	.	...	...	.	...	...	.		.	.
	.	.	.	...	...	.	...	...	.		.	.
	0	0	.	$\frac{1}{y_{rk}}$	...	0	...	$\frac{y_{rj}}{y_{rk}}$	.	1	.	$\frac{\bar{b_r}}{y_{rk}}$
	.	.	.	...	...	...	...	...	.		.	.
	.	.	.	...	...	...	...	...	.		.	.
	.	.	.	...	...	...	...	...	.		.	.
	0	0	.	$-\frac{y_{mk}}{y_{rk}}$	...	0	...	$y_{mj} - \frac{y_{rj}}{y_{rk}}y_{mk}$	.	0	.	$\bar{b_m} - \frac{y_{mk}}{y_{1k}}\bar{b_r}$

Tabla 3: *Después de pivotear*

Interpretación de los elementos de la tabla

	z	$\boldsymbol{x_B}$	$\boldsymbol{x_N}$	**RHS**
z	1	0	$c_B B^{-1} N - c_N$	$c_B B^{-1} b$
$\boldsymbol{x_B}$	0	I	$B^{-1} N$	$B^{-1} b$

Tabla 4: *Esta tabla da toda la información necesaria para desarrollar el método simplex.*

z: Variable de Costo
x_B: Variables básicas
z y x_B están expresados en términos de las variables no básicas x_N, por tanto puede pensarse que las variables no básicas son variables independientes mientras que x_b y z son variables dependientes.

1. $z + 0x_B + (c_B B^{-1} N - c_N) x_N = c_B B^{-1} b$

   $z = c_B B^{-1} b - (c_B B^{-1} N - c_N) x_N$

   $z = c_B B^{-1} b - \sum_{j \in N} (z_j - c_j) x_j$

   $\frac{\partial z}{\partial x_j} = (c_B B^{-1} b)' - [(z_1 - c_1) x_1]' + [(z_2 - c_2) x_2]' + \ldots + [(z_j - c_j) x_j]' + \ldots + [(z_n - c_n) x_n]'$

   $\frac{\partial z}{\partial x_j} = 0 - 0 + 0 + \ldots + (z_j - c_j) + \ldots + 0$

   $\frac{\partial z}{\partial x_j} = -(z_j - c_j) = (c_j - z_j)$

   $(c_B B^{-1} N - c_N) x_N = (c_B B^{-1} \underbrace{[a_1, \ldots, a_j, \ldots, a_n]}_{j \in N} - \underbrace{[c_1, \ldots, c_j, \ldots, c_n]}_{j \in N})$

2. $z + 0x_B + (c_B B^{-1} N - c_N) x_N = c_B B^{-1} b$
   $z = c_B B^{-1} b - (c_B B^{-1} N - c_N) x_N$

   Calculamos
   $(c_B B^{-1} N - c_N) x_N$

$$(c_B B^{-1} N - c_N) x_N = (c_B B^{-1} \underbrace{[a_1, \ldots, a_j, \ldots, a_n]}_{j \in N} - \underbrace{[c_1, \ldots, c_j, \ldots, c_n]}_{j \in N}) \begin{bmatrix} x_1 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{bmatrix}$$

$$([c_B B^{-1} a_1, \ldots, c_B B^{-1} a_j, \ldots, c_B B^{-1} a_n] - [c_1, \ldots, c_j, \ldots, c_n]) \begin{bmatrix} x_1 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{bmatrix}$$

$$([c_B B^{-1} a_1 - c_1, \ldots, c_B B^{-1} a_j - c_j, \ldots, c_B B^{-1} a_n - c_n]) \begin{bmatrix} x_1 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{bmatrix}$$

$(c_B B^{-1} a_1 - c_1) x_1 + \cdots + (c_B B^{-1} a_j - c_j) x_j + \cdots + (c_B B^{-1} a_n - c_n) x_n = \sum_{j \in N} (c_B B^{-1} a_j - c_j) x_j =$

$\sum_{j \in N} (z_j - c_j) x_j$

Por tanto

$z = c_B B^{-1} b - \sum_{j \in N} (z_j - c_j) x_j$

Sea ahora

$\frac{\partial z}{\partial x_j} = (c_B B^{-1} b)' - [(z_1 - c_1) x_1]' + [(z_2 - c_2) x_2]' + \ldots + [(z_j - c_j) x_j]' + \ldots + [(z_n - c_n) x_n]'$

$\frac{\partial z}{\partial x_j} = 0 - 0 + 0 + \ldots + (z_j - c_j) + \ldots + 0$

$\frac{\partial z}{\partial x_j} = -(z_j - c_j) = (c_j - z_j)$

Como $z = c_B B^{-1} - \sum_{j \in N} (z_j - c_j) x_j$ y $\frac{\partial z}{\partial x_j} = -(z_j - c_j)$

Por tanto se debe incrementar x_j si $\frac{\partial z}{\partial x_j} < 0 \Rightarrow -(z_j - c_j) < 0 \Rightarrow (z_j - c_j) > 0$

Con esto terminaremos de comprobar que en un problema de mínimo, $z_j - c_j > 0$
La rapidez de cambio de z como función de una variable no básica cualquiera x_j es $\frac{\partial z}{\partial x_j}$

3. Variables básicas en términos de las variables no básicas

$$x_B = B^{-1}b - B^{-1}Nx_N$$

$$x_B = B^{-1}b - B^{-1}[a_1, \cdots, a_j, \cdots, a_n] \begin{bmatrix} x_1 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{bmatrix}$$

$$x_B = B^{-1}b - [B^{-1}a_1, \cdots, B^{-1}a_j, \cdots, B^{-1}a_n] \begin{bmatrix} x_1 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{bmatrix}$$

$$x_B = B^{-1}b - [B^{-1}a_1x_1 + B^{-1}a_2x_2 + \cdots + B^{-1}a_nx_n] = B^{-1}b - \sum_{j \in N} B^{-1}a_jx_j$$

$$x_B = B^{-1}b - \sum_{j \in N} B^{-1}a_jx_j = B^{-1}b - \sum_{j \in N} y_jx_j = B^{-1}b - [y_1x_1 + \cdots + y_jx_j + \cdots + y_nx_n]$$

$$\frac{\partial x_B}{\partial x_j} = 0 - [0 + \cdots + y_j + \cdots + 0 + \cdots + 0] = -y_j$$

$$\frac{\partial x_B}{\partial x_j} = -y_j$$

Rapidez de cambio de la variable básica como función de la variable no básica x_j

En otras palabras, si x_j se incrementa en una unidad, entonces la i-ésima variable básica x_{Bi} decrece en una cantidad y_{ij} o simplemente $\dfrac{\partial x_B}{\partial x_j} = -y_j$

4. Una columna y_j se puede interpretar como sigue:

$$By_j = a_j \Rightarrow a_j = [a_{B1}, a_{B2}, \cdots, a_{Bi}, \cdots, a_{Bm}] \begin{bmatrix} y_{1j} \\ \vdots \\ x_{ij} \\ \vdots \\ x_{mj} \end{bmatrix} = \sum_{i=1}^{m} a_{Bi} y_{ij}$$

$$a_j = \sum_{i=1}^{m} a_{Bi} y_{ij}$$

a_j se representa como combinación lineal de las columnas básicas requeridas para representar a_j y los y_{ij}

5. Rapidez de cambio de fo con respecto al lado derecho

$$z = c_B B^{-1} b - \sum_{j \in N} (z_j - c_j) x_j$$
$$\frac{\partial z}{\partial b} = c_B B^{-1} b$$

6. Rapidez de cambio de las variables básicas con respecto al lado derecho

$$x_B = B^{-1} b - B^{-1} N x_N$$

$$\frac{\partial x_B}{\partial b} = B^{-1}. \quad En\ particular \quad \frac{\partial x_{Bi}}{\partial b} \quad es\ la\ \ i - ésima\ fila\ de\ B^{-1}$$

$$\frac{\partial x_B}{\partial b_j} \quad es\ la\ j - ésima\ columna\ de \quad B^{-1}\ y\ \frac{\partial x_{Bi}}{\partial b_j} \quad es\ el\ elemento\ (i,j)\ de \quad B^{-1}$$

$$B^{-1} = \begin{bmatrix} & (i,j) & \end{bmatrix} \rightarrow \frac{\partial x_{Bi}}{\partial b}$$
$$\downarrow \qquad \downarrow$$
$$\frac{\partial x_B}{\partial b_j} \qquad \frac{\partial x_{Bi}}{\partial b_j}$$

9.1.3.4 Solución Inicial y Convergencia

En muchos casos es difícil disponer de una solución básica factible inicial. Esto lleva a realizar un trabajo previo antes de iniciar el método simplex. algunos procedimientos son: El Método de las dos fases, el método de de penalizacion o de gran M (Este método tiene el inconveniente de no ser tan práctico en la implementación) y un método de poca utilización que es el de una variable artificial.

La Solución Básica Factible Inicial

El método simplex empieza con una Solución Básica Factible (SPB) y llega una SPB mejorada cuando se alcanza el punto óptimo o bien hasta que se verifica el no acotamiento de la función objetivo. Sin embargo, para iniciar el método simplex se debe disponer de una base B con $x_B = B^{-1}b \geq 0$. (Lo ideal es que B sea I)

Caso Fácil

$$\begin{array}{ll} \text{Minimizar} & cx \\ \text{s.a} & Ax \geq b, \quad A_{mxn}, \quad rg(A) = m \\ & x \geq 0 \end{array}$$

Agregando variables de holgura

$$\begin{array}{ll} \text{Minimizar} & cx + 0x_n \\ \text{s.a} & Ax + x_h = b \\ & x \geq 0, \quad x_h \geq 0, \quad b \geq \mathbf{0_n} \end{array}$$

En este caso:

$$Ax + x_h = b \Rightarrow [\tilde{A}, I] \begin{bmatrix} x \\ x_h \end{bmatrix} = b$$
$$B = I, \quad N = \tilde{A}, \quad x_B = b, \quad x_N = \mathbf{0}$$
$$A = [\tilde{A}, B]$$
$$A = [\tilde{A}, B]$$

Caso Difícil

En muchas ocasiones encontrar una SPB inicial no es tan fácil como el caso descrito anteriormente, por ejemplo:

1. $Ax \leq b$, $x \geq 0$, pero $b \ngeq 0$, en este caso cuando introducimos el vector de holgura x_h, no

podemos tomar $x_N = 0$, pues $x_s = b \ngeq 0$ y viola la restricción de no negatividad de x_B.

2. Otra situación ocurre cuando las restricciones son de la forma $Ax \geq b$, $x \geq 0$, $b \geq 0$. Cuando hacemos $Ax - x_h = b$, nos queda una matríz $-I$, por tanto no podemos hacer $x_B = B^{-1}b \geq 0$, ya que $B = -I$.

Variables Artificiales

Si después de manipular las restricciones e introducir las x_h obtenemos $Ax = b$, $x \geq 0$, con A_{mxn}, y $b \geq$, pero A no contiene una submatriz identidad. En este caso se recurrirá a variables artificiales para obtener una SPB y después se usará el método simplex para eliminar estas variables artificiales.

Minimizar	$c^t x$	Minimizar	$c^t x$
Sujeto a	$Ax \lessgtr b$	s.a	$Ax + x_a = b$
	$x \geq 0$		$x \geq 0, x_a \geq 0$

Este es un problema diferente al original ya

$$A = [\tilde{A}, I]$$
$$x_B = B^{-1}b = b = x_a, x_N \geq 0$$
$$x = [\tfrac{x_B}{x_N}] = [\tfrac{x_a}{x_N}]$$

Y las variables artificiales x_a no son legítimas en el problema.

Pero $x = [\tfrac{x_a}{x_N}] \geq 0$ es una SPB del nuevo problema y nos servirá para iniciar el método simplex para resolver el problema general.

Formalizando lo Anterior

Supóngase que se combinan las restricciones añadiendo un vector artificial x_a, lo cual nos da el sistema $Ax + x_a = b$, $x \geq 0$, $x_a \geq 0$. Nótese que, por construcción, ahora se tiene una matriz identidad correspondiente al vector artificial. Esto da de inmediato una spb del nuevo sistema, a saber, $x_a = b$,y $x_N = 0$. Aunque ahoara se tiene una spb inicial y se puede aplicar el método simplex, tenemos una dificultad, el problema ya no es el mismo, sino que es otro.

Para regresar al problema original, debe obligarse a que estas variables artificiales se hagan cero, pues $Ax = b \Rightarrow Ax + x_a = b$, lo que obliga $x_a = 0$. En otras palabras las x_a son solo una herramienta que se usa para empezar el método simplex, pero debe garantizarse que estas variables eventualmente se hagan cero.

Las x_h y x_a son diferentes, ya que las x_h, se introducen para poner el problema en forma de igualdad. Las x_a no son legítimas, pero se introducen para facilitar el inicio del método simplex. Las x_a deben hacerse cero para lograr la factiblidad en el problema original.

Problema original

$$\begin{array}{l} \text{Minimizar} \quad c^t x \\ \text{s.a} \quad Ax \lessgtr b \\ x \geq 0 \end{array}$$

Usando x_a, tenemos otro problema

$$\begin{array}{l} \text{Minimizar} \quad c^t x \\ \text{s.a} \quad Ax + x_a = b \\ x \geq 0, x_a \geq 0 \\ x = [\frac{x_B}{x_N}] = [\frac{x_a}{x_N}] = [\frac{B^{-1}b}{\mathbf{0}}] \end{array}$$

Para el Problema Lineal (PL) original debemos hacer $x_a = 0$, o sea tener una

$$\begin{array}{l} x = [\frac{x_B}{x_N}] = [\frac{B^{-1}b \geq 0}{\mathbf{0}}] \\ Para\ que\ Ax + x_a = b \\ Ax + \mathbf{0} = b \\ Ax = b \end{array}$$

9.1.3.5 El Método de Dos Fases

Fase I. Minimizar la suma de las variables artificiales sujeta a las restricciones $Ax + x_a = b$, $x \geq 0$ y $x_a \geq 0$. Si el problema original tiene una solución factible, entonces el valor óptimo de este problema es cero con todas las variables atificiales iguales a cero.

A medida que la variables artificiales se hecen cero, salen de la base y, en su lugar entran variables legítimas. Eventualmente todas las variables artificiales salen de la base (Aunque éste no siempre es el caso, porque se puede tener una variable artificial en la base en el nivel cero).

Asi, pues la base consiste de variables legítimas. En otras palabras, se obtiene una solución básica factible (SPB) del problema original $Ax = b$, $x \geq 0$ y el método simplex se puede iniciar con el objetivo original cx.

Por otra parte, si después de resolver este problema, se tiene una variable artificial positiva, entonces el problema original no tiene soluciones factibles.

$$\begin{cases} Se\ reducen\ las\ variables\ artificiales \\ a\ ceroo\ problema\ original\ no\ tiene\ soluciones \\ factibles \end{cases} \Rightarrow Fase\ II\ :\ Se\ minimiza\ la\ f.o.original.$$

Fase I

$$\begin{array}{ll} \text{Minimizar} & 1x_a \\ \text{s.a} & Ax + x_a = b \\ & x, x_a \geq 0 \end{array}$$

El PL inicia con las SPB $x_a = x_B = b$, $x_N = 0$. si al alcanzar optimalidad $x_a \neq 0$, entonces el proceso termina, en este caso el PL no tiene soluciones factibles.

Fase II

$$\begin{array}{ll} \text{Minimizar} & cx = c_B x_B + c_N x_N \\ \text{s.a} & x_B + B^{-1}Nx_N = B^{-1}b \\ & x_B, x_N \geq 0 \end{array}$$

Este PL inicia con la spb $x_B = B^{-1}b$ y $x_N = 0$

La solución óptima de este problema es la solución óptima del problema original.

9.1.3.6 El Método de Penalización

Una forma de eliminar variables artificiales consiste en asignar coeficientes para estas variables en la función objetivo original, en forma tal que hagan que su presencia en la base, en un nivel positivo, sea muy poco atractiva desde el punto de vista de la función objetivo. Por ejemplo, supóngase que se desea resolver el siguiente problema de PL, en el que $\mathbf{b} \geq 0$.

$$\begin{array}{ll} \text{Minimizar} & cx \\ \text{s.a} & Ax = b \\ & x \geq 0 \end{array}$$

Si no se conoce una base conveniente, se introduce el vector artificial, x_a, lo cual conduce al siguiente sistema:

$$\begin{array}{l} Ax + x_a = b \\ x, x_a \geq 0 \end{array}$$

La solución básica factible inicial está dada por $x_a = b$ y $x = 0$. Para reflejar la inconveniencia de un vector artificial distinto de cero, la función objetivo se modifica de tal forma que se pague un castigo o pena muy alta cuando se tenga una solucion de este tipo. Más específicamente, considérese el siguiente problema:

$$\begin{array}{ll} \text{Minimizar} & cx + M_1 x_a \\ \text{s.a} & Ax + x_a = b \\ & x, x_a \geq 0 \end{array}$$

En donde M_1 es una constante positiva muy grande, el término $M_1 x_a$ se puede interpretar como una pena o multa que se debe pagar por cualquier solución con $x_a \neq 0$

Aunque la solución inicial $x = 0$, $x_a = b$, es factible para las nuevas restricciones, tiene un valor objetivo muy poco atractivo, a saber, $M_1 b$. Por lo tanto, el mismo método simplex tratará de sacar a las variables artificiales fuera de las base, y después, continuará hasta encontrar la solución óptima del problema original.

9.1.3.7 Regla Lexicográfica (Regla de Salida)

$$I_0 = \left\{ r : \frac{\bar{b}_r}{y_{rk}} = \min_{1 \leq i \leq m} \left\{ \frac{\bar{b}_i}{y_{ik}} : y_{ik} > 0 \right\} \right\}$$

Si I_0 consiste de un solo elemento, a saber $I_0 = \{r\}$, entonces x_{Br} sale de la base. En caso contrario, se forma I_1 como sigue:

$$I_1 = \left\{ r : \frac{y_{r1}}{y_{rk}} = \min_{i \in I_0} \left\{ \frac{y_{i1}}{y_{ik}} \right\} \right\}$$

Si I_1 consiste de un solo elemento, a saber $I_1 = \{r\}$, entonces x_{Br} sale de la base. En caso contrario, se forma I_2. En general, dado I_{j-11}, el conjunto I_j se forma como sigue:

$$I_j = \left\{ r : \frac{y_{rj}}{y_{rk}} = \min_{i \in I_{j-1}} \left\{ \frac{y_{ij}}{y_{ik}} \right\} \right\}$$

Eventualmente, para algún $j < m$, I_j consistirá solo de un elemento. Si $I_j = \{r\}$, entonces x_{Br} sale de la base.

Nota. Primero se emplea la prueba de la razón mínima como criterio de salida. Si esta prueba da un índice único, entonces la variable correspondiente sale de la base. En caso de un empate,

se trata de romperlo reemplazando el lado derecho en el cálculo de la razón mínima, por la primera columna y_1 y usando únicamente filas correspondiente al empate. Si el empate sigue sin romperse, se usa la segunda columna, y asi sucesivamente. Al alcanzar la columna m o antes, el empate debera romperse, porque de no ser así, se tienen dos filas de la matríz $B^{-1} = (y_1, y_2, ..., y_m)$ que son proporcionales. Sin embargo, esto es imposible, por la independencia lineal de las filas de B_{-1}

9.1.3.8 El Método Simplex Revisado

El método simplex revisado es un procedimiento sistemático para implementar los pasos del método simplex en un arreglo más pequeño, ahorrando así, espacio de almacenamiento.

Pasos del método simplex:

- $x_B = B^{-1}b = \bar{b}$
  $x_N = 0$
  $Z = c_B B^{-1} b = c_B \bar{b}$

- $\forall j \in N$ calcular $z_j - c_j = c_B B^{-1} a_j - c_j$
  Sea $z_k - c_k$ el máximo de los $z_j - c_j$
  Si $z_k - c_k \leq 0$ stop solución óptima
  $z_k - c_k > 0$ ir al paso 3

- Calcular $y_k = B^{-1} a_k$
  Si $y_k \leq 0$ stop la solución óptima es no acotada.
  Calcular

  $$x_k = \frac{\bar{b}_r}{y_{rk}} = \min_{1 \leq i \leq m} \left\{ \frac{\bar{b}_i}{y_{ik}} : y_{ik} > 0 \right\}$$

Se actualiza B, reemplazando a_{Br} con a_k y se regresa al paso 1.

Todo lo anterior se puede operar con un arreglo más pequeño. El cálculo de los $z_j - c_j$ se hace afuera aprovechando que se conoce $c_B B^{-1}$ y así determinar si el problema termina o si se introduce una nueva variable.

9.2 El Problema del Agente Viajero (TSP)

Existen múltiples formulaciones distintas del TSP. En este apartado se van a revisar algunas de las más utilizadas.

El problema del TSP puede ser descrito según la teoría de grafos de la siguiente manera: Sea $G = (V, A)$ un grafo completo, donde $V = \{1, 2, 3, ..., n\}$ es el conjunto de vértices y A es el conjunto de aristas. Los vértices $i = \{2, 3, 4, ..., n\}$ se corresponden con los nodos a visitar y el vértice 1 es el nodo de origen y destino. A cada arco (i, j) se le asocia un valor no negativo c_{ij}, que representa el coste de viajar del vértice i al j. El uso de los arcos (i, i) no está permitido, por lo que se impone $c_{ii} = \infty$ para todo $i \in V$. Si G es un grafo dirigido, la matriz de costes C es asimétrica mientras que, si $c_{ij} = c_{ji}$ para todo $(i, j) \in A$, la matriz de costes será simétrica y el problema recibirá el nombre de TSP simétrico (STSP). En ese caso, el conjunto A se sustituye por un conjunto E de arcos no dirigidos (i, j) tales que $i < j$.

El objetivo del Problema del Agente Viajero es encontrar una ruta que, comenzando y terminando en una ciudad determinada, en este caso denotada por la ciudad i, pase una sola vez por cada una de las ciudades y minimice la distancia recorrida. Si se definen las variables dicotómicas de decisión x_{ij} para todo $(i, j) \in A$, de forma que tomen el valor 1 si el arco (i, j) forma parte de la solución y 0 en otro caso,

$$x_{ij} = \begin{cases} 1 & \text{si } j = 0, \\ 0 & \text{si } j = 1, 2, \ldots, n, \end{cases}$$

entonces se tiene que el problema de programación lineal asociado al Problema del Agente Viajero consiste en minimizar la siguiente función objetivo:

$$\sum_{ij} c_{ij} x_{ij}$$

sujeto a las siguientes restricciones:

$$\sum_{j \in \delta^-} x_{ji} = 1, \quad \forall i \in V,$$

$$\sum_{j \in \delta^+} x_{ij} = 1, \quad \forall i \in V,$$

donde

$$\delta^- = \{a = (j, i) \in A\}; \quad \delta^+ = \{a = (i, j) \in A\}$$

La primera restricción se refiere a que solo un arco puede entrar en cada vértice, mientras que la segunda se refiere a que solo un arco puede salir de cada nodo.

Estas restricciones son necesarias pero no suficientes, pues pueden dar lugar a un subtour, como se puede observar en la siguiente figura:

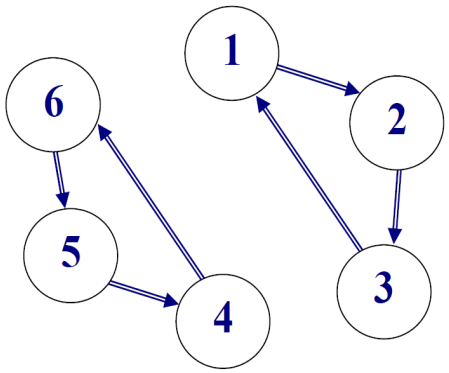


Figura 14: Subtour

Obsérvese que $x_{12} = x_{23} = x_{31} = x_{54} = x_{46} = x_{65} = 1$, por lo que no se viola ninguna de las restricciones.

Por lo tanto, se debe incluir alguna restricción más de ruptura de subtour. En la figura anterior se puede observar que en el subconjunto $\{1, 2, 3\}$ hay 3 arcos que unen los nodos entre sí. Si se limita este número de arcos a 2, se evitarían situaciones como esta. Para poder modelizar estas últimas restricciones, es necesario nuevos conjuntos:

$$\forall\, W \subset V,\ A(W) = \{a = (i,j) \in A : i,j \in W\}$$

$$\delta^-(W) = \{a = (i,j) \in A : i \notin W, j \in W\}$$

$$\delta^+(W) = \{a = (i,j) \in A : i \in W, j \notin W\}$$

Así, las restricciones de ruptura de subtour pueden escribirse de la siguiente manera:

$$\sum_{(i,j)\in A(W)} x_{ij} \leq \mid W \mid -1,\ \forall\, W \subset V$$

Que es equivalente a:

$$\sum_{i\in W, j\notin W} x_{ij} \geqslant 1,\ \forall\, W \subset V$$

Esta restricción indica que para todo subconjunto de los nodos debe haber al menos un arco que "salga" del subconjunto. Otra forma de evitar la formación de subtours es a través de nuevas variables de decisión. De esta forma, se definen las variables $u_i,\ \forall i \in V$, que representan el lugar de la

secuencia en el que se visita el nodo i. Para el nodo origen 1, prefijamos el valor de u_0 en 1, pues es el primer nodo que se debe visitar. Para el resto de vértices, estas variables toman un valor entre 2 y n. Para construir el problema de programación lineal se deben añadir a las dos primeras restricciones, la siguiente:

$$u_i - u_j \leq (n-1)(1-x_{ij}) - 1,\ \forall (i,j) \in A, j \geq 1$$

La interpretación de esta restricción es la siguiente: si el agente viajero va de i directamente a j, entonces x_{ij} valdrá 1 y $u_i - u_j \leq -1$. Dado que se visita antes el nodo i que el j, u_i tomará un valor menor que u_j y, por tanto, su diferencia valdrá (-1), cumpliéndose así la inecuación. Si el agente viajero no va de i a j, x_{ij} tomará el valor 0, con lo que $u_i - u_j \leq n-2$. En este caso, no se tiene información adicional acerca de u_i y u_j, por lo que nos centramos en el extremo: si i se visita antes que j, la máxima diferencia entre u_i y u_j será $2-n$, mientras que si j se visita antes que i , $u_i - u_j$ tomará un valor mínimo de $n-2$. En todos los casos, por tanto, se cumple que $u_i - u_j \leq n-2$. Obviamente, de existir subtours estas variables no podrán tomar valores que cumplieran esta restricción, pues no se puede establecer qué vértices se visitan antes y cuáles se visitan después.

En la primera de las formulaciones se requieren $2^n + 2n - 2$ restricciones ($2^n - 2$ por las de ruptura de subtour, n por la restricción de "entrada" y n por la restricción de "salida" de cada nodo) y $n \cdot (n-1)$ variables dicotómicas. Por el contrario, para la segunda formulación se necesitan $2^n - n + 2$ restricciones ($(n-1)\cdot(n-2)$ por las restricciones asociadas a las variables u_i, y $2n$ por los otros dos tipos de restricciones), $n \cdot (n-1)$ variables dicotómicas y $(n-1)$ variables continuas. Si se comparan ambas formulaciones, en el primero de los casos hay más restricciones pero menos variables que en el segundo caso. En cada caso particular y, dependiendo de la forma de resolución de este problema de programación lineal entera, merecerá la pena utilizar una u otra formulación.

No obstante, se puede demostrar que la formulación con las restricciones de ruptura de subtour domina a la formulación con las u_i.

Por último, veamos la primera formulación para el caso del TSP simétrico. De nuevo, las variables de decisión toman el valor 1 si el arco que une y pertenece al tour solución y 0 en otro caso. La formulación es la que sigue:

$$Min \sum_{ij} c_{ij} x_{ij}$$

$$sa \sum_{j \in \delta(i)} x_{ij} = 2,\ \forall i \in V$$

$$\sum_{(i,j) \in E(W)} x_{ij} \leq \mid W \mid -1,\ \forall W \subset V,\ \frac{n}{2} \leq \mid W \mid \geq 3$$

$$x_{ij} \in \{0,1\},\ \forall (i,j) \in E$$

siendo $\delta_i = \{e \in E : e = (i,j)\ \ e = (j,i)\};\ E(W) = \{(i,j) \in E : i,j \in W\}$
y $\delta(S) = \{e = (i,j) \in\ E : (i \in S, j \notin S)(i \notin S, j \in S)\}$

Como ocurre con el caso asimétrico, la segunda restricción puede sustituirse por:

$$\sum_{(i,j)\in\delta(W)} x_{ij} \geq 2$$

9.3 Complejidad Computacional del TSP

La solución más directa para resolver el Problema del Agente Viajero es sin duda, intentar todas las permutaciones (combinaciones ordenadas) y ver cuál de estas es la menor (usando una Búsqueda de Fuerza Bruta). El tiempo de ejecución es un factor polinómico de orden $O((n-1)!)$, el factorial del número de ciudades, esta solución es impracticable para dado solamente 21 ciudades. Se tendrían que realizar $20! = 2\ 432902\ 008176\ 640000$ evaluaciones, casi dos trillones y medio de permutaciones. Incluso para una computadora la tarea sería muy complicada, por no decir muy tardada. Una de las mejores aplicaciones de la Programación dinámica es el algoritmo Held–Karp que resuelve el problema en $O(n^2 2^2)$.

Y para un mayor número de ciudades, enumerar las soluciones y escoger la mejor, resulta ser una hazaña que podría durar años, siglos y hasta miles de millones de años. Por ejemplo:

$$\begin{aligned}
10 \text{ ciudades:} &\quad \approx (10^{5.5}) \text{ posibilidades} \\
100 \text{ ciudades:} &\quad \approx (10^{156}) \text{ posibilidades} \\
1000 \text{ ciudades:} &\quad \approx (10^{2565}) \text{ posibilidades} \\
33810 \text{ ciudades:} &\quad \approx (10^{138441}) \text{ posibilidades} \\
10 \text{ ciudades:} &\quad \approx (10^{5.5}) \text{ posibilidades} \\
\text{Edad del Universo:} &\quad \approx (10^{18}) \text{ segundos} \\
\text{Número de Átomos en el Universo:} &\quad < (10^{100})
\end{aligned}$$

Sin embargo usando métodos más inteligentes es posible encontrar la solución en menos tiempo. Una solución exacta para 15 112 pueblos alemanes desde TSPLIB fue encontrada en 2001 usando el método de planos cortantes propuesto por George Dantzig, Ray Fulkerson, y Selmer M. Johnson en 1954, basados en la programación lineal. Los cálculos fueron realizados por una red de 110 procesadores ubicados en la Universidad Rice y en la Universidad de Princeton. El tiempo total de cálculo fue equivalente a 22.6 años en un Procesador alpha de 500 MHz.

En mayo de 2004, el TSP de visitar todos los 24 978 poblados en Suecia fue resuelto: un recorrido de tamaño aproximado de 72 500 kilómetros fue encontrado y se probó que no existía un camino menor.

En marzo de 2005, el Problema del Agente Viajero de visitar todos los 33 810 puntos en una tabla de circuitos fue resuelto usando Concorde TSP Solver: un recorrido de 66 048 945 unidades fue encontrado y se probó que no existía un recorrido menor. El cálculo tomo aproximadamente 15.7 años - CPU. En abril de 2006, una instancia con 85 900 puntos fue resuelta usando "Concorde TSP Solver", tomando 136 años- CPU (Applegate et al., 2006).

Lo anterior puede dar una idea de qué significa complejidad computacional. Pues su objetivo fundamental es clasificar los problemas de acuerdo a su manejabilidad, es decir, ¿es posible obtener una solución tomando él o los algoritmos más eficientes para resolverlos?, o bien, se quiere determinar las respuestas a las siguientes preguntas:

- ¿Qué tan manejable es el problema?
- Si el problema es manejable, ¿es eficiente el algoritmo?

En general, los distintos grados de complejidad son subjetivos (varían considerablemente de acuerdo al modelo computacional, a los recursos disponibles, a las variantes de las estructuras de datos, etc.). Por lo tanto, un objetivo primario del estudio de la complejidad es definir cuáles problemas son tratables, y cuáles no.

Los problemas de acuerdo con su complejidad de solución se clasifican en clase P, clase NP, clase NP-completo y clase NP-Hard.

- **Clase P.** La clase de complejidad de los problemas que pueden ser resueltos en tiempo polinomial calculado a partir de la entrada por una máquina de Turing determinista es llamada P. Una máquina de Turing no es una máquina física sino un mecanismo lógico por medio del cual el cálculo puede descomponerse en iteraciones de operaciones concretas extremadamente simples (controladas por un "programa").
- **Clase NP.** Cuando los problemas sólo se pueden resolver usando una máquina de Turing no-determinista, se dice que pertenecen a la clase NP, el cual es el acrónimo en inglés de Polinómico No determinista (Non-Deterministic Polynomial-time).
- **Clase NP-Completo.** Un problema de decisión C es NP-Completo sí es un problema NP y todo problema de NP se puede transformar polinomialmente en él.

- **Clase NP-Hard.** Cuando se prueba que un problema de optimización combinatoria en su versión problema de decisión (combinatorio binario), pertenece a la clase NP-completa, entonces la versión optimización es NP-Hard.

En la siguiente figura se resume lo anterior:

Figura 15: Complejidad Computacional

La complejidad computacional no está dentro de los objetivos de esta investigación. Solo diremos que el Problema del Agente Viajero (TSP) es un NP-Hard.

Aunque se ha logrado resolver de manera exacta problemas simétricos hasta con 85 900 ciudades la demanda de recursos computacionales se incrementa considerablemente al aumentar el tamaño del problema, esto ha llevado a buscar alternativas de solución a problemas de tipo NP, lo cual ha conducido al desarrollo de métodos no exactos (Heurísticos y Metaheurísticos) que dan una respuesta a estos problemas.

9.4 Algunos Métodos de Solución del TSP

Las formas tradicionales de tratar este problema intentan encontrar una solución exacta, cuando las dimensiones del problema son relativamente pequeñas; se han utilizado algoritmos heurísticos, que proporcionan buenas soluciones, aunque no se puede garantizar siempre que sean las óptimas; o bien, se divide el problema en subproblemas reduciendo las dimensiones del problema original y posiblemente su dificultad.

Un desarrollo más elaborado, es la aplicación de algunos algoritmos metaheurísticos como el Tabu Search (Búsqueda Tabú) o el Scatter Search (Búsqueda Dispersa).

9.4.1 Algoritmos Exactos

La forma más directa de encontrar la solución exacta sería formular todas las permutaciones, siendo el número de soluciones posibles $n!$. Esta posibilidad se convierte en impracticable a medida que aumenta el número de variables. Otras opciones son: utilizar un proceso de **branch-and-bound** (Ramificación y Acotamiento o Poda), o bien implementar algoritmos basados en programación lineal, con un buen funcionamiento respecto al número de vértices o nodos; incluso, podrían combinarse ambas opciones.

La técnica **Branch and Bound**, en particular, se suele interpretar como un árbol de soluciones, donde cada rama nos lleva a una posible solución posterior a la actual. La característica principal de este método es que el algoritmo se encarga de detectar en qué ramificación las soluciones dadas ya no están siendo óptimas, para "podar" esa rama del árbol y no continuar malgastando recursos y procesos en casos que se alejan de la solución óptima.

El pseudocódigo del Algoritmo de Ramificación y Acotamiento es el siguiente:

Algoritmo 1. Pseudocódigo del Branch and Bound

```
Función RyP {
P = Hijos(x,k)
while ( no vacio (P) )
    x(k) = extraer(P)
    if esFactible (x,k) y G(x,k) < optimo
        si esSolucion (x)
            Almacenar(x)
        else
            RyP(x,k+1)
```

Figura 16: Pseudocódigo Ramificación y Acotamiento

Donde:

- **G(x)** es la función de estimación del algoritmo.
- **P** es el vector permutación.

- **esFactible** es la función que considera si la propuesta es válida.
- **esSolucion** es la función que comprueba si se satisface el objetivo.
- **optimo** es el valor de la función a optimizar evaluado sobre la mejor solución encontrada hasta el momento.

9.4.2 Algoritmos Heurísticos

Este término deriva de la palabra griega heuriskein que significa encontrar o descubrir y se usa en el ámbito de la optimización para describir una clase de algoritmos de resolución de problemas

En el lenguaje coloquial, optimizar significa poco más que mejorar; sin embargo, en el contexto científico la optimización es el proceso de tratar de encontrar la mejor solución posible para un determinado problema. En un problema de optimización existen diferentes soluciones y un criterio para discriminar entre ellas. De forma más precisa, estos problemas se pueden expresar como encontrar el valor de unas variables de decisión para los que una determinada función objetivo alcanza su valor máximo o mínimo. El valor de las variables en ocasiones está sujeto a unas restricciones.

En Díaz y otros (1996) se recogen hasta ocho definiciones diferentes de algoritmo heurístico, entre las que se destaca la siguiente:

Un método heurístico es un procedimiento para resolver un problema de optimización bien definido mediante una aproximación intuitiva, en la que la estructura del problema se utiliza de forma inteligente para obtener una buena solución.

En contraposición a los métodos exactos que proporcionan una solución óptima del problema, los métodos heurísticos se limitan a proporcionar una buena solución no necesariamente óptima. Lógicamente, el tiempo invertido por un método exacto para encontrar la solución óptima de un problema difícil, si es que existe tal método, es de un orden de magnitud muy superior al del heurístico (pudiendo llegar a ser tan grande en muchos casos, que sea inaplicable).

Los algoritmos heurísticos diseñados en esta investigación fueron el Nearest Neighbor (Vecino Más Cercano) y el Greedy Randomized Adaptive Search Procedures-GRASP (Procedimientos de Búsqueda basados en funciones “Voraces o Codiciosos”). También se crearon, aunque con menos éxito en las soluciones, un Algoritmo de Permutaciones, Genético y Path Relinking (Reencaminamiento). A continuación, se abordan los dos primeros.

9.4.2.1 Heurístico del Vecino Más Cercano

Atendiendo a la definición anterior de algoritmo heurístico se puede afirmar que el Vecino Más Cercano es el heurístico más sencillo e intuitivo para resolver el TSP. Este algoritmo trata de construir un ciclo o tour hamiltoniano de bajo coste basándose en el vértice más próximo a uno dado. Su planteamiento se debe Rosenkrantz, Stearns y Lewis (1977) y su código, en una versión standard, es el siguiente:

Algoritmo 2. Pseudocódigo del Vecino Más Cercano

Inicialización

Seleccionar un vértice j al azar

Hacer $t = j$ y $W = V \setminus \{j\}$

while $(W \neq \phi)$

Tomar $j \in W / c_{tj} = \min\{c_{ti} / i \in W$

Conectar t a j

Hacer $W = W \setminus \{j\}$ y $t = j$

Figura 17: Pseudocódigo del Vecino Más Cercano

Este procedimiento realiza un número de operaciones de orden $O(n^2)$. Si seguimos la evolución del algoritmo al construir la solución de un ejemplo dado, veremos que comienza muy bien, seleccionando aristas de bajo coste. Sin embargo, al final del proceso probablemente quedarán vértices cuya conexión obligará a introducir aristas de coste elevado. Esto es lo que se conoce como miopía del procedimiento, ya que, en una iteración escoge la mejor opción disponible sin tomar en cuenta que esto puede obligar a realizar malas elecciones en iteraciones posteriores, por tanto se consideran algoritmos greedy.

El algoritmo tal y como aparece puede ser programado en unas pocas líneas de código. Sin embargo una implementación directa será muy lenta al ejecutarse sobre ejemplos de gran tamaño (10000 vértices). Así pues, incluso para un heurístico tan sencillo como éste, es importante pensar en la eficiencia y velocidad de su implementación.

9.4.2.2 Heurístico GRASP

Los métodos GRASP fueron desarrollados al final de la década de los 80 con el objetivo inicial de resolver problemas de Set Covering (Cubrimientos de Conjuntos, Feo y Resende, 1989). El término GRASP fue introducido por Feo y Resende (1995) como una nueva técnica metaheurística de propósito general.

GRASP es un procedimiento de multi-arranque en donde cada paso consiste en una Fase Consructiva y una de Mejora. En la fase de construcción se aplica un procedimiento heurístico constructivo para obtener una buena solución inicial. Esta solución se mejora en la segunda fase mediante un algoritmo de búsqueda local. La mejor de todas las soluciones examinadas se guarda como resultado final.

Los elementos de este procedimiento son: fase de construcción, de mejora y actualización.

En la fase de construcción se construye iterativamente una solución posible, considerando un elemento en cada paso. En cada iteración la elección del próximo elemento para ser añadido a la solución parcial viene determinada por una función greedy. Esta función mide el beneficio de añadir cada uno de los elementos según la función objetivo y elegir la mejor. Nótese que esta medida es miope en el sentido que no tiene en cuenta qué ocurrirá en iteraciones sucesivas al realizar una elección, sino únicamente en esta iteración.

Se dice que el heurístico greedy se adapta porque en cada iteración se actualizan los beneficios obtenidos al añadir el elemento seleccionado a la solución parcial. Es decir, la evaluación que se tenga de añadir un determinado elemento a la solución en la iteración, no coincidirá necesariamente con la que se tenga en la iteración.

Este heurístico es aleatorizado porque no selecciona el mejor candidato según la función greedy adaptada sino que, con el objeto de diversificar y no repetir soluciones en dos construcciones diferentes, se construye un lista con los mejores candidatos de entre los que se toma uno al azar.

Al igual que ocurre en muchos métodos, las soluciones generadas por la fase de construcción de GRASP no suelen ser óptimos locales. Dado que la fase inicial no garantiza la optimalidad local respecto a la estructura de entorno en la que se esté trabajando (hay selecciones aleatorias), se aplica un procedimiento de búsqueda local como postprocesamiento para mejorar la solución obtenida.

En la fase de mejora se suele aplicar un procedimiento de intercambio simple con el objeto de no

emplear mucho tiempo en esta mejora. GRASP se basa en realizar multiples iteraciones y quedarse con la mejor, por lo que no es especialmente beneficioso para el método el detenerse demasiado en mejorar una solución dada.

El siguiente esquema muestra el funcionamiento global del algoritmo:

Algoritmo 3. Pseudocódigo del GRASP

while (condición de parada)

Fase Constructiva

Seleccionar una lista de elementos candidatos.
Considerar una Lista Restringida de los mejores Candidatos.
Seleccionar un elemento aleatoriamente de la Lista.

Fase de Mejora

Realizar un proceso de búsqueda local a partir de la solución construida hasta que no se pueda mejorar más.

Actualización

Si la solución obtenida mejora a la mejor almacenada, actualizarla.

Figura 18: Pseudocódigo GRASP

El realizar muchas iteraciones GRASP es una forma de realizar un muestreo del espacio de soluciones. Basándose en las observaciones empíricas, se ve que la distribución de la muestra generalmente tiene un valor en promedio que es inferior al obtenido por un procedimiento determinista, sin embargo, la mejor de las soluciones encontradas frecuentemente supera a la del procedimiento determinista con una alta probabilidad.

Las implementaciones GRASP generalmente son robustas en el sentido de que es difícil el encontrar ejemplos patológicos en donde el método funcione arbitrariamente mal.

9.4.3 Algoritmos Metaheurísticos

En los últimos años han aparecido una serie de métodos bajo el nombre de Metaheurísticos con el propósito de obtener mejores resultados que los alcanzados por los heurísticos tradicionales. El término metaheurístico fue introducido por Fred Glover (Colorado University-1986). En esta investigación se utilizará la acepción de heurísticos para referirnos a los métodos clásicos en contraposición a la de metaheurísticos que se reserva para los más recientes y complejos. En algunos textos se puede encontrar la expresión "heurísticos modernos" refiriéndose a los metaheurísticos (Reeves, 1995). Los profesores Osman y Kelly (1995) introducen la siguiente definición:

Los procedimientos metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son efectivos. Los metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos.

Los algoritmos metaheurísticos diseñados en este trabajo fueron el Tabu Search (Búsqueda Tabú) y el Scatter Search (Búsqueda Dispersa). A continuación, se exponen ambos.

9.4.3.1 Algoritmo Tabu Search

El término Búsqueda Tabú (Tabu Search) fue introducido en 1986 por Fred Glover en el mismo artículo que introdujo el término metaheurística (Glover, 1986). Los principios fundamentales de la búsqueda tabú fueron elaborados en una serie de artículos de finales de los años 80 y principios de los 90, que fueron luego unificados en el libro "Tabu Search" en 1997. El destacado éxito de la búsqueda tabú para resolver problemas de optimización difíciles, especialmente aquellos que surgen en aplicaciones del mundo real, ha causado una explosión de nuevas aplicaciones durante los últimos años, que aparecen resumidas en (Glover et al, 2006).

Búsqueda tabú es una técnica para resolver problemas combinatorios de gran dificultad que está basada en principios generales de Inteligencia Artificial. En esencia es un metaheurístico que puede ser utilizado para guiar cualquier procedimiento de búsqueda local en la búsqueda agresiva del óptimo del problema. Por agresiva se refiere a la estrategia de evitar que la búsqueda quede "atrapada" en un óptimo local que no sea global.

Para tal efecto, la Búsqueda Tabú toma de la inteligencia artificial el concepto de memoria y lo implementa mediante estructuras simples con el objetivo de dirigir la búsqueda teniendo en cuenta

la historia de ésta. Es decir, el procedimiento trata de extraer información de lo sucedido y actuar en consecuencia. En este sentido puede decirse que hay un cierto aprendizaje y que la búsqueda es inteligente. El principio del Tabu Search podría resumirse como:

Es mejor una mala decisión basada en información que una buena decisión al azar, ya que, en un sistema que emplea memoria, una mala elección basada en una estrategia proporcionará claves útiles para continuar la búsqueda. Una buena elección fruto del azar no proporcionará ninguna información para posteriores acciones.

En resumen, la búsqueda tabú es una combinación de búsqueda local con un mecanismo de memoria a corto plazo. Siendo sus elementos claves los siguientes:

- Restricciones Tabú: restringir la búsqueda al clasificar ciertos movimientos como prohibidos (tabú), para evitar caer en soluciones recientemente generadas.
- Criterio de aspiración: liberar la búsqueda por medio de una función de memoria a corto plazo (olvido estratégico).

En la siguiente tabla se muestra el funcionamiento del algoritmo:

Algoritmo 4. Pseudocódigo del Tabu Search

1. Selecciona un estado $x \in X$ inicial y sea $x^* := x$, $k = 0$ (contador de iteración) y $T := \phi$.
2. Si $S(x) - T = \phi$ ve a 4. Sino $k := k + 1$ y selecciona $s_k \in (S(x) - T)$ tal que $s_k(x) = OPTIMO\{s(x) : s \in (S(x) - T)\}$.
3. Sea $x := s_k(x)$. Si $c(x) < c(x^*)$ (Donde x^* es la mejor solución encontrada hasta el momento), sea $x^* := x$.
4. Si se agotó el número de interacciones o si $S(x) - T = \phi$; entonces para, Sino, actualiza T (añade el movimiento actual a la lista tabú y posiblemente elimina el elemento más viejo) y regresa a 2.

Figura 19: Pseudocódigo Tabú Search

La siguiente figura muestra un ejemplo sencillo del funcionamiento de Búsqueda Tabú:

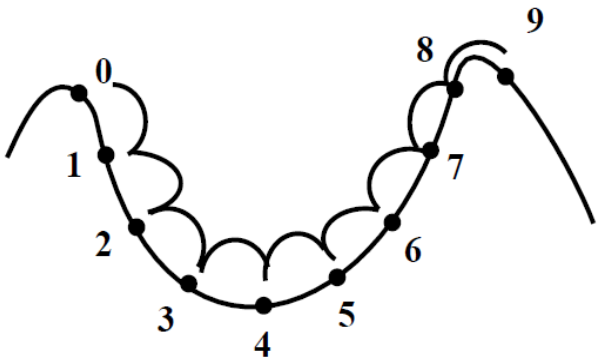


Figura 20: Búsqueda Tabú

Supongamos que en cada punto solo pueden hacerse dos movimientos, hacia una ciudad con un número anterior o a una ciudad con un número posterior y supongamos que nuestra lista tabú es de tamaño 3.

Supongamos que se inicia en el punto marcado con el número 1 en la figura anterior. De ahí se pueden hacer dos movimientos (hacia 0 y hacia 2). Se elige el mejor (hacia 2), esto se denota como mov(1, 2) y se actualiza la lista tabú (inicialmente vacía). Se registra el movimiento inverso en la lista (mov(2, 1)) para evitarlo en el futuro.

De ahí la siguiente ciudad a moverse es la 3 (el mejor y el permitido dada la lista tabú actual) y después al 4, con lo que la lista tabú se llena con tres elementos:

$$\{mov(4,3), mov(3,2), mov(2,1)\}$$

El siguiente movimiento es peor en la función objetivo (mov(4, 5)), pero es el mejor dentro de los permitidos por la lista tabú y por el esquema de vecindad que se ha utilizado. También provoca que se elimine el elemento más viejo que tiene la lista tabú (por estar llena). La lista tabú queda como:

$$\{mov(5,4), mov(4,3), mov(3,2)\}$$

Este proceso continua hasta que finalmente se pueda salir del mínimo local al moverse de la ciudad 8 a la 9.

Características:

- Las soluciones dependen de como se actualiza T.
- No hay condición de óptimo local.
- Se busca la "mejor" solución en cada paso (función OPTIMO), en lugar de alguna opción que mejore la solución.
- OPTIMO puede ser:

$$c(s_k(x)) = \textit{mínimo}\{c(S(x)) : s \in S(x) - T\}$$

- OPTIMO da la mejor solución o la menos peor sujeta a la lista tabú.

En principio, se podría tomar alguna solución que mejore (en caso de que sea difícil encontrar la mejor), pero normalmente se sigue la estrategia más agresiva. La razón principal es que los óptimos locales no presentan problemas.

Normalmente la lista tabú se implementa como una lista circular, añadiendo elementos en la posición 1 y eliminando los que sobrepasen la posición (para una t fija).

Una forma efectiva de implementar la lista T sería:

$$T = \{s^{-1} : s = s_h \quad para \quad h > k - t\}$$

donde k es el número de iteración y s^{-1} es el movimiento inverso de s . Por lo que el paso de actualización de T sería:

$$T := T - s_{k-t}^{-1} + s_k^{-1}$$

En general, lo que se trata de evitar es caer en ciudades o nodos solución previos. Esto no quiere decir que se tenga que escoger una t muy grande.

En la práctica T no toma la forma anterior:

- s^{-1} Previene un conjunto más grande de movidas,
- Por consideraciones de memoria es deseable guardar solo un subconjunto de atributos que caracterizan el movimiento.

Bajo estas condiciones, la lista tabú representa una colección de movidas c_h.

Cuando la lista $S - T = \emptyset$; se pueden eliminar los elementos más viejos de T para permitir continuar con el proceso.

9.4.3.2 Algoritmo Scatter Search

La primera descripción de este método fue publicada en 1977 por Fred Glover donde establece los principios de la "Búsqueda Dispersa". En este primer artículo se determina que la búsqueda dispersa realiza una exploración sistemática sobre una serie de buenas soluciones llamadas conjunto de referencia. Los siguientes comentarios resumen los principales aspectos de este trabajo:

- El método se centra en combinar dos o más soluciones del conjunto de referencia. La combinación de más de dos soluciones tiene como objetivo el generar centroides.
- Generar soluciones en la línea que unen dos dadas se considera una forma reducida del método.
- Al combinar se deben de seleccionar pesos apropiados y no tomar valores al azar.
- Se deben de realizar combinaciones no convexas de las soluciones.
- La distribución de los puntos se considera importante y deben de tomarse dispersos.

El Scatter Search o Búsqueda Dispersa es un método de evolución diferente de los algoritmos genéticos, opera sobre un conjunto pequeño de soluciones y usa limitadamente la aleatoriedad para poder diversificar cuando busca una solución global óptima. Su estructura es flexible, permite desarrollar implementaciones alternativas con variación en el grado de sofisticación. La búsqueda dispersa es un procedimiento que consiste de cinco métodos:

- Un Método de Generación de Diversificación para generar una colección de diversas soluciones de pruebas, usando una o más de estas soluciones de prueba arbitrariamente como una entrada o inicio.
- Un Método de Mejora para transformar una solución de prueba en una o más soluciones de pruebas intensificadas (no se requiere que las soluciones de entrada o de salida sean posibles, aunque la solución de salida es típicamente posible. Si la solución de entrada no mejora como resultado de la aplicación de este método, el aumento de la solución es considerado a ser igual a la solución de entrada).
- Un Método de Actualización del Conjunto de Referencia para construir y mantener un conjunto de referencia consistente de las b "mejores" soluciones halladas (donde el valor de b es típicamente pequeño, por ejemplo no más de 20), es organizado para proveer un acceso eficiente a las otras partes del procedimiento solución. Varios criterios alternativos pueden ser usados para adicionar y eliminar soluciones al conjunto de referencia.

- Un Método de Generación de Subconjuntos para operar sobre el conjunto de referencia y producir un subconjunto de soluciones como una base para crear combinación de soluciones. El método más común de generación de subconjuntos es el de generar todos los pares de las soluciones del conjunto de referencia (es decir, todos los subconjuntos de tamaño 2).
- Un Método de Combinación de Soluciones para transformar un subconjunto dado de soluciones producidas por el método de generación de subconjuntos en una o más soluciones combinadas. Este método es análogo al de operador de cruce en el algoritmo genético pero este es capaz de combinar dos o más soluciones.

Resumen de la Metodología del Scatter Search

La metodología del Scatter Search es muy flexible, ya que cada uno de los elementos puede ser implementado en una variedad de maneras y grado de sofisticación. Su algoritmo puede resumirse de la siguiente manera:

Algoritmo 5. Pseudocódigo del Scatter Search

1. Comenzar con $P = \phi$. Utilizar el método de generación de la diversificación para construir una solución x. Si $x \notin P$ entonces añadir x a P (es decir, $P = P \cup x$), en caso contrario, descartar x. Repetir esta etapa hasta que P tenga un tamaño prefijado, $|P| = Psize$.
2. Construir el conjunto de referencia Re $fSet = \{x^1, \ldots, x^b\}$ con soluciones diversas de P. (con las mejores $b/2$ soluciones de P y las $b/2$ soluciones de P más diversas a las ya incluidas.)
3. Evaluar las soluciones en Re $fSet$ y ordenarlas de mejor a peor respecto a la función objetivo, tal que x es la mejor solución y x^b la peor.
4. Hacer $NuevaSolución = TRUE$
5. **While (** $NuevaSolución$ **) do**
   Generar nuevos subconjuntos, el cual consiste de todos los pares de soluciones en Re $fSet$ que incluye al menos una nueva solución.
   Hacer $NuevaSolución = FALSE$
   While ($NuevosSubconjuntos \neq \phi$ **) do**
      Seleccionar el nuevo subconjunto s en $NuevosSubconjuntos$.
      Aplicar el método de combinación de la solución a s para obtener una o más nuevas soluciones x.
      If (x **no está en** Re $fSet$ **y** $f(x) < f(x^b)$ **) then**
         Hacer $x^b = x$ y reordenar Re $fSet$
         Hacer $NuevaSolución = TRUE$
      end if
      Eliminar s de $NuevosSubconjuntos$
   end While
6. **end While**

Figura 21: Pseudocódigo Scatter Search

10 METODOLOGÍA

Para alcanzar los propósitos de este trabajo se realizó una revisión del estado del arte, centrándonos en la profundida y alcance con que ha sido abordado el problema del TSP (Traveling Salesman Problem).

Durante esta revisión se tomaron en cuenta las investigaciones con grandes aportes teóricos y prácticos sobre el Problema del Agente Viajero, así como los artículos clásicos y más importantes publicados recientemente según la literatura relevante del TSP. Pérez (2011) propone el estudio de las diez publicaciones más citadas, entre 2006-2010, que tuvieron al TSP como eje central (tema o problema principal de prueba). De igual manera se tomaron en cuenta las tendencias para avanzar alrededor del conocimiento actual sobre el TSP y para mover las puntas de conocimiento hacia otras esferas.

Con la revisión del estado del arte, se pudo encontrar la trayectoria de científicos que han hecho importantes aportes a este problema con el desarrollo de los llamados heurísticos modernos o metaheurísticos. El primero de ellos es el profesor Fred Glover de la Universidad de Colorado, con un total de 57930 citas a la fecha según el Google Académico. En segundo lugar el profesor de Investigación de Operaciones, Manuel Laguna, también de la Universidad de Colorado con 21779 citas. En tercer lugar se encuentra el profesor de Investigación de Operaciones y Estadísticas Rafael Martí de la Universidad de Valencia con 8166 citas. Y siguiendo esta línea, llegamos a encontrar a mi tutor de Tesis Doctoral el Doctor Antonio Parajón Guevara con aproximadamente 200 referencias en el índice de citas. Esta ha sido en esencia la bibliografía consultada para el desarrollo de este trabajo.

El diseño metodológico de esta investigación se puede apreciar en la siguiente figura. Inicia con el diseño de algoritmos heurísticos y metaheurísticos para resolver el TSP, continua con la implementación en lenguaje C++ y finaliza con la comparación de los resultados con las mejores soluciones de la TSPLIB (Traveling Salesman Problem Library) publicada desde 1991 por Gerhard Reinelt de la Universidad de Heidelberg, Alemania.

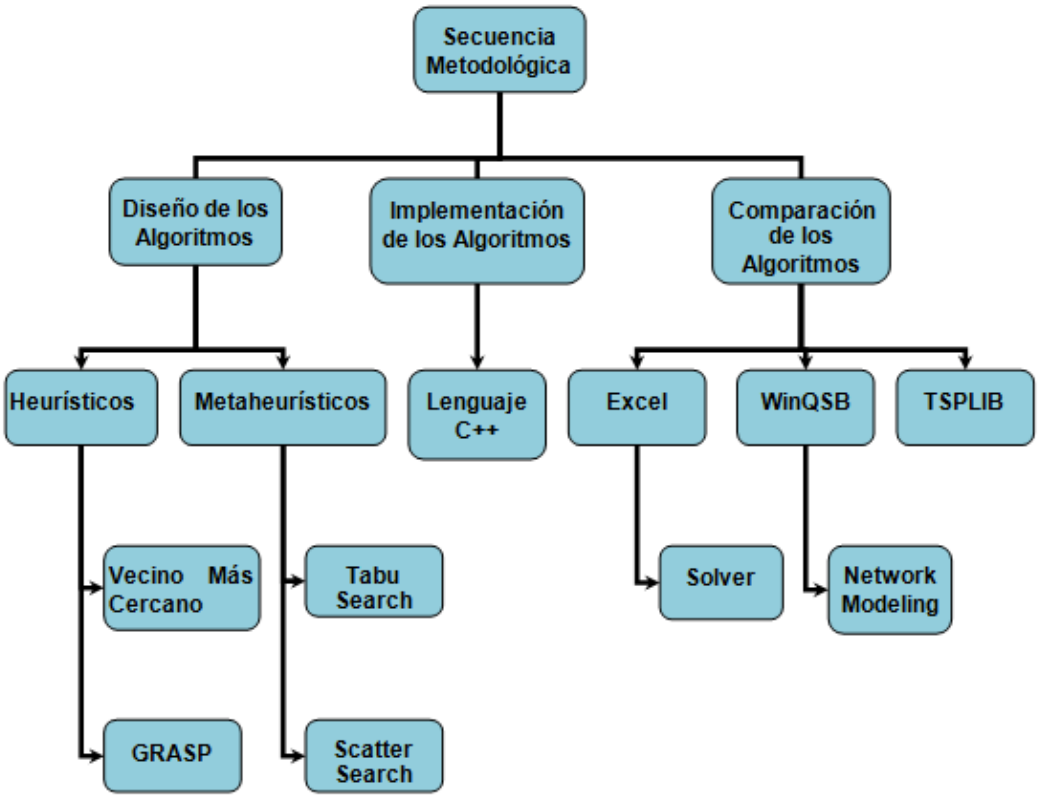


Figura 22: Secuencia Metodólogica

10.1 Diseño de Algoritmos Heurísticos para el TSP

Para el diseño de los algoritmos heurístico del Vecino Más Cercano y GRASP se inició trabajando con los diagramas de flujos para representar gráficamente cómo funciona cada método. Cada paso del proceso se representa por un símbolo diferente que contiene una breve descripción de cada paso.

Los símbolos gráficos del flujo están unidos entre sí con flechas que indican la dirección de cada paso del método. Esto ofrece una descripción visual de las etapas implicadas, mostrando la relación secuencial entre ellas, facilitando la rápida comprensión de cada acción y su relación con las demás, el flujo de la información y resultados parciales, las ramas del proceso, la existencia de bucles respetitivos, etc., facilitando la comprensión del problema a programar.

Dado que estos diagramas representan la secuencia lógica o los pasos que tenemos que efectuar para realizar una tarea, resultan ser una excelente herramienta para resolver problemas y para hacer programas informáticos.

Una vez que se ha desarrollado el diagrama de flujo, se procede a introducir las órdenes del lenguaje de programación para realizar las tareas que se especifican en el esquema. Tales instrucciones se pueden encontrar en los pseudocódigos propuestos para cada algoritmo en el apartado anterior de esta investigación.

10.1.1 Algoritmo del Vecino Más Cercano

Este algoritmo heurístico es el procedimiento más intuitivo para el el Problema del Agente Viajero. En el código diseñado, lo primero que se hace es tomar el número de nodos de cada instancia desde un archivo que hemos llamado de lectura. Luego se selecciona un nodo al azar y tomando como criterio visitar al que tenga la distancia mínima se completa el tour, visitando una vez cada vértice. El siguiente diagrama de flujo general muestra en forma resumida el proceso.

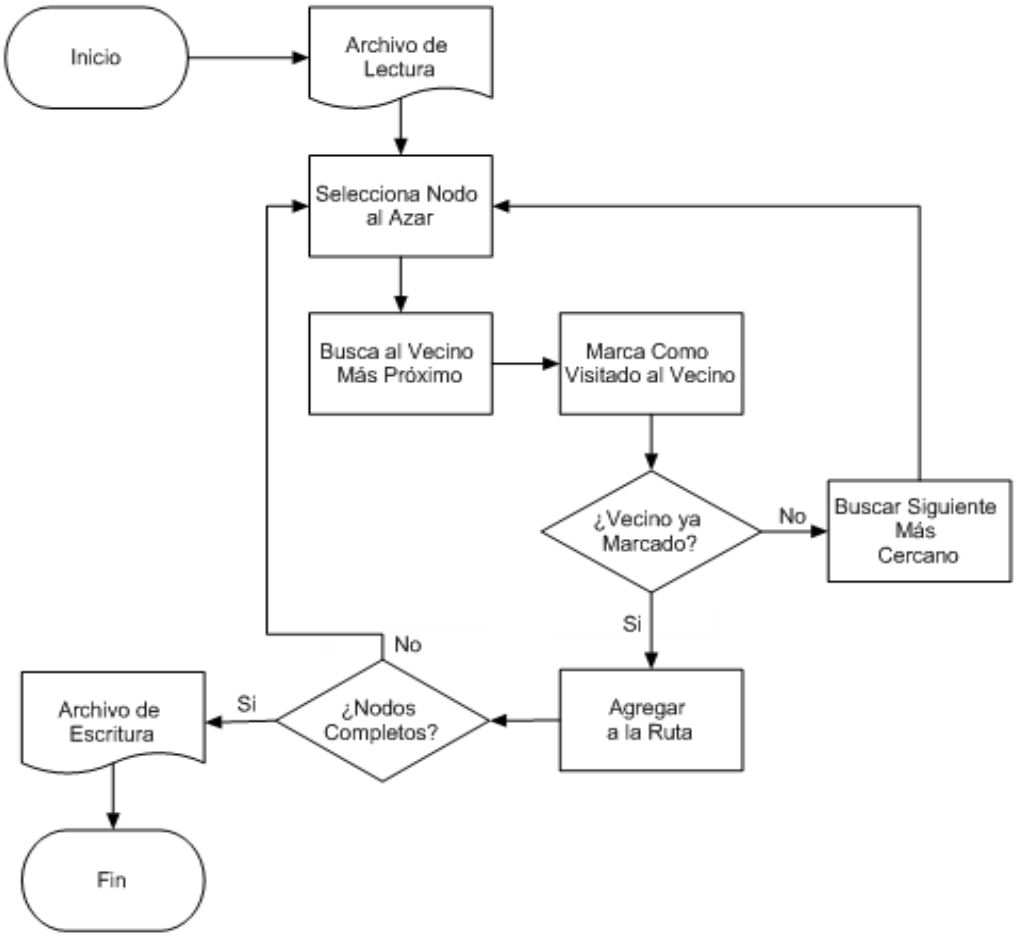


Figura 23: Diagrama de Flujo Vecino Más Cercano

10.1.2 Algoritmo GRASP

Según se ha descrito anteriormente, el GRASP consta de dos fases distintas: en la primera se construye la solución inicial o de entrada, y en la segunda se trata de mejorar la solución construida, mediante una metodología de búsqueda local. En este caso el diagrama de flujo tiene la forma siguiente:

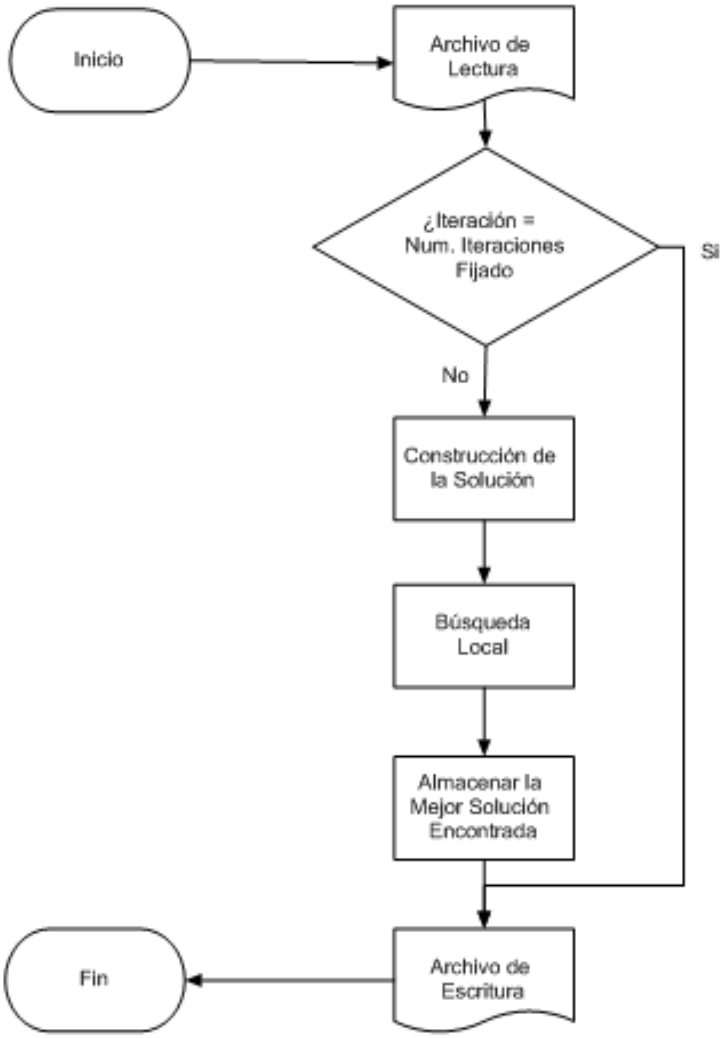


Figura 24: Diagrama de Flujo GRASP

10.2 Diseño de Algoritmos Metaheurísticos para el TSP

Para el diseño de los algoritmos metaheurísticos se trabajó en forma análoga a la descrita en el caso de los heurísticos.

10.2.1 Tabu Search o Búsqueda Tabú

En el algoritmo básico del Tabu Search se hace uso de la memoria a corto plazo, la cual conforma el nucleo de la búsqueda tabú, según se planteó en el pseudocódigo presentado en el marco teórico. En versiones mejoradas del algoritmo se puede incluir una estructura de memoria de largo plazo, con estrategias de intensificación de la búsqueda en un área promisoria del espacio de soluciones y diversificación hacia otras zonas no exploradas. El diagrama de flujo se presenta a continuación:

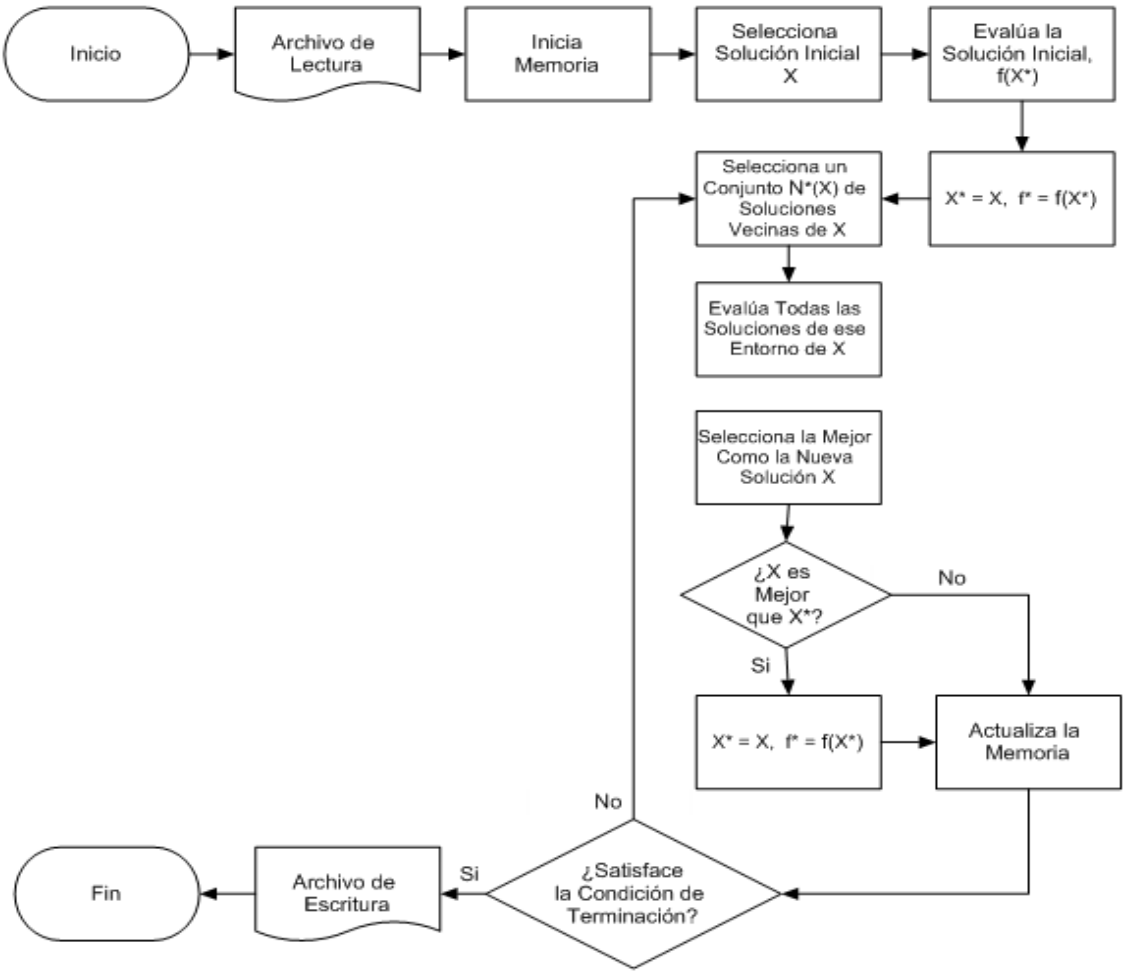


Figura 25: Diagrama de Flujo Tabu Search

10.2.2 Scatter Search o Búsqueda Dispersa

Según se dijo en el marco teórico, el Scatter Search está basado en la combinación de un grupo de soluciones que se encuentran almacenadas en un conjunto de referencia, denominado RefSet. El objetivo de las combinaciones es generar centroides, centros geométricos, a fin de obtener nuevas soluciones de mayor calidad. A continuación se describe su funcionamiento con el digrama de flujo que fue utilzado para diseño de su código en C++:

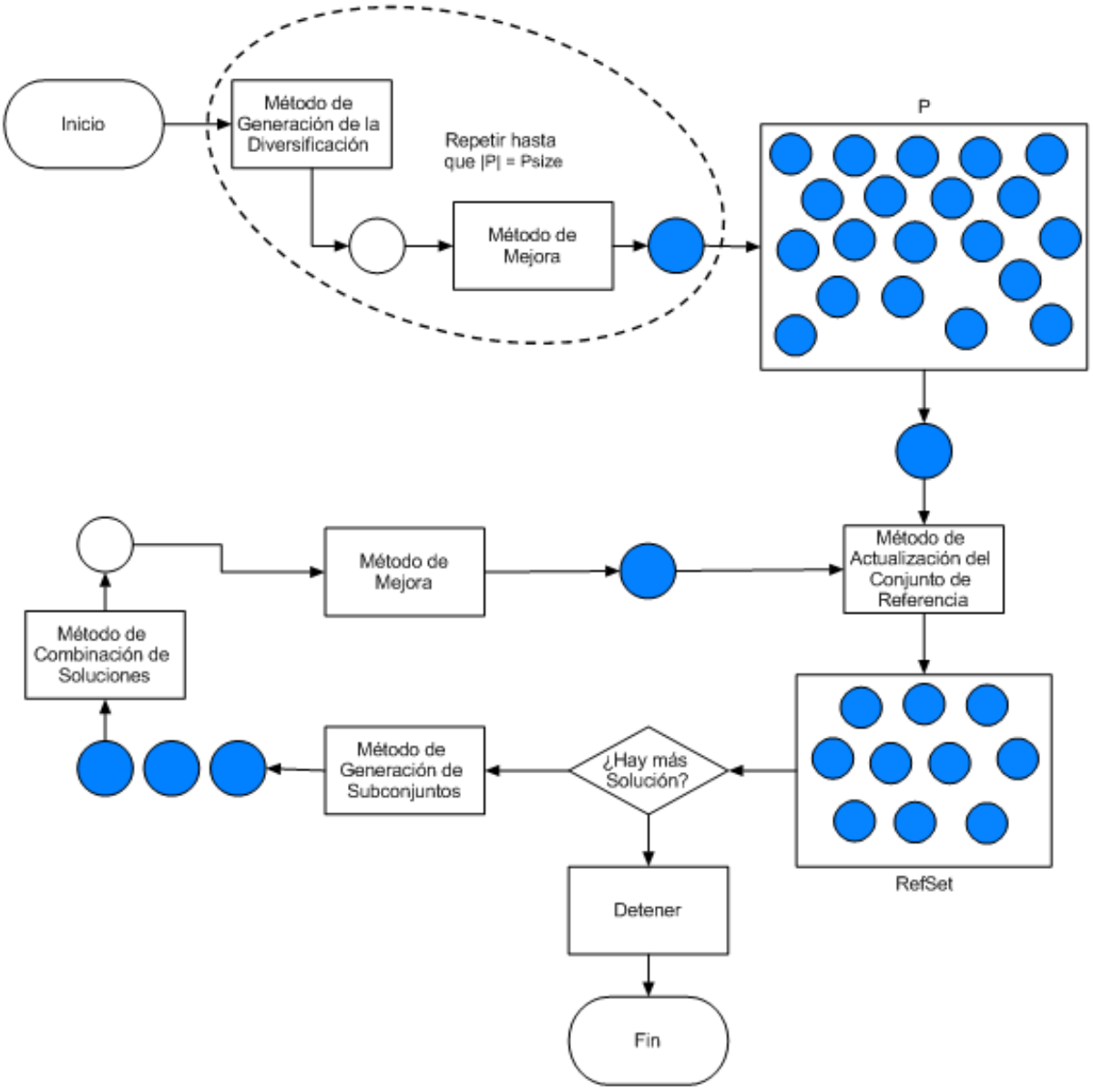


Figura 26: Diagrama de Flujo Scatter Search

10.3 Implementación de Algoritmos para el TSP

Los algoritmos heurísticos y metaheurísticos diseñados para resolver el TSP se implementaron en el Microsoft Visual Studio 2008 que es un entorno de desarrollo integrado (IDE). Este soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP; al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc. En esta investigación se trabajó con el lenguaje de programación C++.

El procedimiento de trabajo inicia creando un PROYECTO para el algoritmo. Luego se deben grabar en él las instancias del TSP porque el código diseñado leerá las matrices simétricas correspondientes. A continuación se presenta una imagen del entorno Visual Studio:

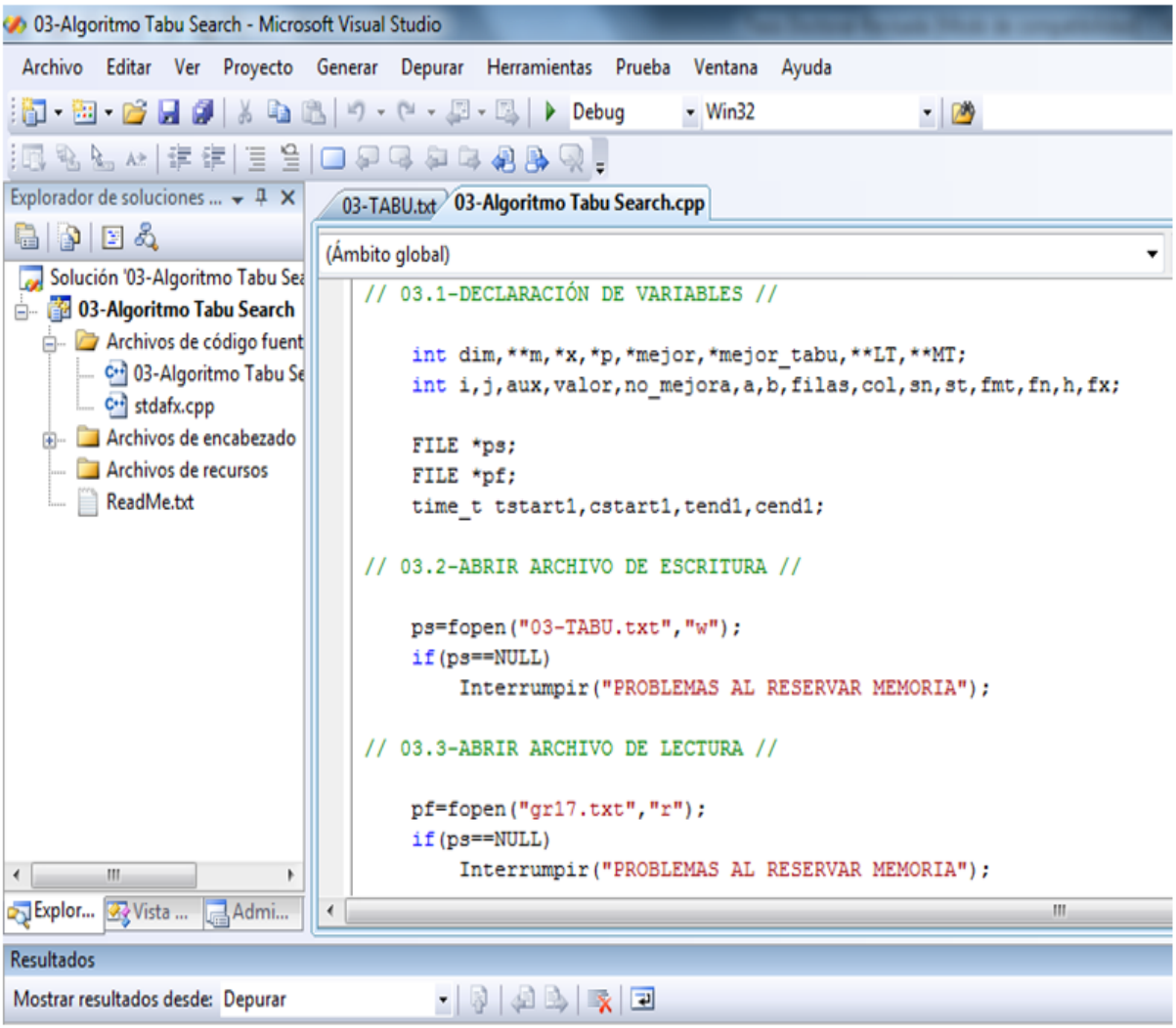


Figura 27: Código en lenguaje C++

10.4 Comparación con la TSPLIB

La TSPLIB es una biblioteca virtual de instancias para probar la calidad de algoritmos diseñados para el TSP (y problemas relacionados) de varias fuentes y de varios tipos. Es una referencia para decir si un algoritmo es bueno o no.

Los científicos utilizan la TSPLIB como un filtro para probar la eficiencia de los algoritmos diseñados en sus investigaciones. En la actualidad contiene 112 instancias, como se muestra en la siguiente tabla:

INSTANCIAS PARA EL TSP (TSPLIB)					
NO	INSTANCIAS	NO	INSTANCIAS	NO	INSTANCIAS
1	a280 : 2579	39	gr48 : 5046	77	pr299 : 48191
2	ali535 : 202339	40	gr96 : 55209	78	pr439 : 107217
3	att48 : 10628	41	gr120 : 6942	79	pr1002 : 259045
4	att532 : 27686	42	gr137 : 69853	80	pr2392 : 378032
5	bayg29 : 1610	43	gr202 : 40160	81	rat99 : 1211
6	bays29 : 2020	44	gr229 : 134602	82	rat195 : 2323
7	berlin52 : 7542	45	gr431 : 171414	83	rat575 : 6773
8	bier127 : 118282	46	gr666 : 294358	84	rat783 : 8806
9	brazil58 : 25395	47	hk48 : 11461	85	rd100 : 7910
10	brd14051 : 469385	48	kroA100 : 21282	86	rd400 : 15281
11	brg180 : 1950	49	kroB100 : 22141	87	rl1304 : 252948
12	burma14 : 3323	50	kroC100 : 20749	88	rl1323 : 270199
13	ch130 : 6110	51	kroD100 : 21294	89	rl1889 : 316536
14	ch150 : 6528	52	kroE100 : 22068	90	rl5915 : 565530
15	d198 : 15780	53	kroA150 : 26524	91	rl5934 : 556045
16	d493 : 35002	54	kroB150 : 26130	92	rl11849 : 923288
17	d657 : 48912	55	kroA200 : 29368	93	si175 : 21407
18	d1291 : 50801	56	kroB200 : 29437	94	si535 : 48450
19	d1655 : 62128	57	lin105 : 14379	95	si1032 : 92650
20	d2103 : 80450	58	lin318 : 42029	96	st70 : 675
21	d15112 : 1573084	59	linhp318 : 41345	97	swiss42 : 1273
22	d18512 : 645238	60	nrw1379 : 56638	98	ts225 : 126643
23	dantzig42 : 699	61	p654 : 34643	99	tsp225 : 3916
24	dsj1000 : 18659688 (EUC_2D)	62	pa561 : 2763	100	u159 : 42080
25	dsj1000 : 18660188 (CEIL_2D)	63	pcb442 : 50778	101	u574 : 36905
26	eil51 : 426	64	pcb1173 : 56892	102	u724 : 41910
27	eil76 : 538	65	pcb3038 : 137694	103	u1060 : 224094

Figura 28: Instancias del TSP

28	eil101 : 629	**66**	pla7397 : 23260728	**104**	u1432 : 152970
29	fl417 : 11861	**67**	pla33810 : 66048945	**105**	u1817 : 57201
30	fl1400 : 20127	**68**	pla85900 : 142382641	**106**	u2152 : 64253
31	fl1577 : 22249	**69**	pr76 : 108159	**107**	u2319 : 234256
32	fl3795 : 28772	**70**	pr107 : 44303	**108**	ulysses16 : 6859
33	fnl4461 : 182566	**71**	pr124 : 59030	**109**	ulysses22 : 7013
34	fri26 : 937	**72**	pr136 : 96772	**110**	usa13509 : 19982859
35	gil262 : 2378	**73**	pr144 : 58537	**111**	vm1084 : 239297
36	gr17 : 2085	**74**	pr152 : 73682	**112**	vm1748 : 336556
37	gr21 : 2707	**75**	pr226 : 80369		
38	gr24 : 1272	**76**	pr264 : 49135		

Figura 29: Instancias del TSP

En esta investigación se utilizaron 41 instancias de tamaño pequeño, mediano y grande para efecturar las comparaciones de los resultados obtenidos con los algortimos heurísticos y metaheurísticos diseñados para resolver el TSP, en contraposición con las soluciones dadas por los softwares Solver y WinQSB.

10.5 Cronograma de la Investigación

Fecha	Actividad a realizar	Responsables	Observación
Enero a Diciembre, 2016	Desarrollo de tesis doctoral y diseño de algoritmos para el TSP.	José Jesús Mendoza Casanova y Dr. Antonio Parajón.	Tutorías del Dr. Antonio Parajón.
Enero, 2017	Elaboración del protocolo de tesis doctoral	José Jesús Mendoza Casanova y Dr. Antonio Parajón	Tutorías del Dr. Antonio Parajón. Taller de tesis #1
Marzo, 2017	Presentación de protocolo de tesis doctoral	José Jesús Mendoza Casanova	Taller de tesis #2
Abril, 2017	Entrega del Protocolo de tesis doctoral	José Jesús Mendoza Casanova	Tutorías del Dr. Antonio Parajón. Declaración de intenciones
Mayo, 2017	Defensa del Protocolo de tesis doctoral	José Jesús Mendoza Casanova	
Agosto, 2017	Entrega de tesis doctoral	José Jesús Mendoza Casanova	Incluye sugerencias al protocolo
Septiembre, 2017	Defensa pública de tesis doctoral escrita.	José Jesús Mendoza Casanova	

Figura 30: Cronograma

11 RESULTADOS Y ANÁLISIS

11.1 Comparación con Solver y WinQSB

En esta parte de la investigación se presentan los resultados obtenidos para 12 instancias de tamaño medio de la TSPLIB. En la siguiente tabla se muestran los resultados obtenidos con el Solver de Excel, WinQSB, Vecino Más Cercano, GRASP y el Tabú Search:

FUNCIÓN OBJETIVO							
	TSPLIB		SOFTWARES		HEURÍSTICOS		METAHEURÍSTICO
NO	INSTANCIA	MEJOR SOLUCIÓN	SOLVER	WINQSB	VECINO 1000-it	GRASP 100000-it	TABU 200-it
1	brazil58	25395	47918	29380	27384	28607	27243
2	dantzig42	699	699	794	864	802	853
3	fri26	937	953	1046	965	937	937
4	gr17	2085	2090	2157	2178	2085	2085
5	gr21	2707	2803	3119	3003	2707	3031
6	gr24	1272	1388	1372	1553	1355	1490
7	gr48	5046	6557	5845	5840	5968	5800
8	gr120	6942	29261	7822	8438	9836	8280
9	hk48	11461	16829	13015	12137	13837	12002
10	si175	21407	26361		22000	33292	22043
11	swiss42	1273	1864	1481	1437	1427	1466
12	tsp8	34	34	34	34	34	34

Óptimo Alcanzado

Figura 31: Función objetivo

En primer lugar, se puede observar que con cada método se ha obtenido el óptimo del “tsp8”, y esto se debe a que se trata de una instancia pequeña con solamente ocho nodos. También se puede apreciar que el Solver, GRASP y el Tabu encontraron la solución óptima para otras cuatro instancias. No obstante, es necesario señalar las debilidades de algunos de ellos.

En el caso del software WinQSB, se puede apreciar que no tiene solución para la instancia “si175”, y esto ocurre porque no es posible introducir en él una matriz simétrica de más de 120 ciudades. Pero aquí no se está hablando de lo difícil con respecto al tiempo que podría ser entrar por teclado una matriz de 175 nodos, sino de la incapacidad para guardar y operar grandes cantidades de datos.

Una instancia de tamaño mediano se puede introcucir fácilmente en WinQSB. Si se tiene a mano dicha instancia en un achivo de texto, se puede copiar y trasladarla a Excel donde su usan los comandos DATOS y Texto en Columnas para obtener la matriz correspondiente. Esta última se puede copiar y pegar en WinQSB para resolverla. Eso sería imposible de realizar con problemas grandes como el "pla85900". Véase en la siguiente figura el caso del “gr17”:

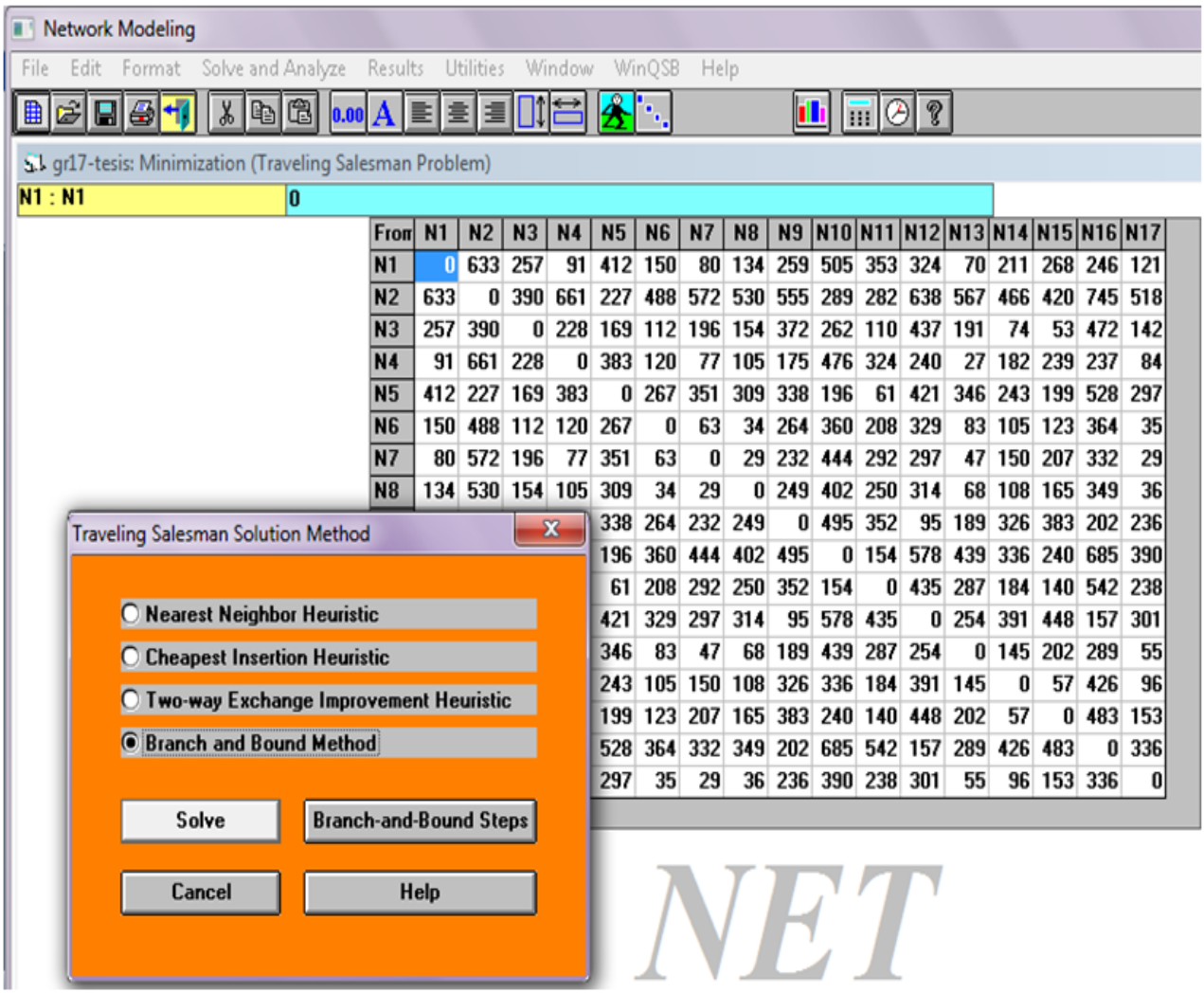
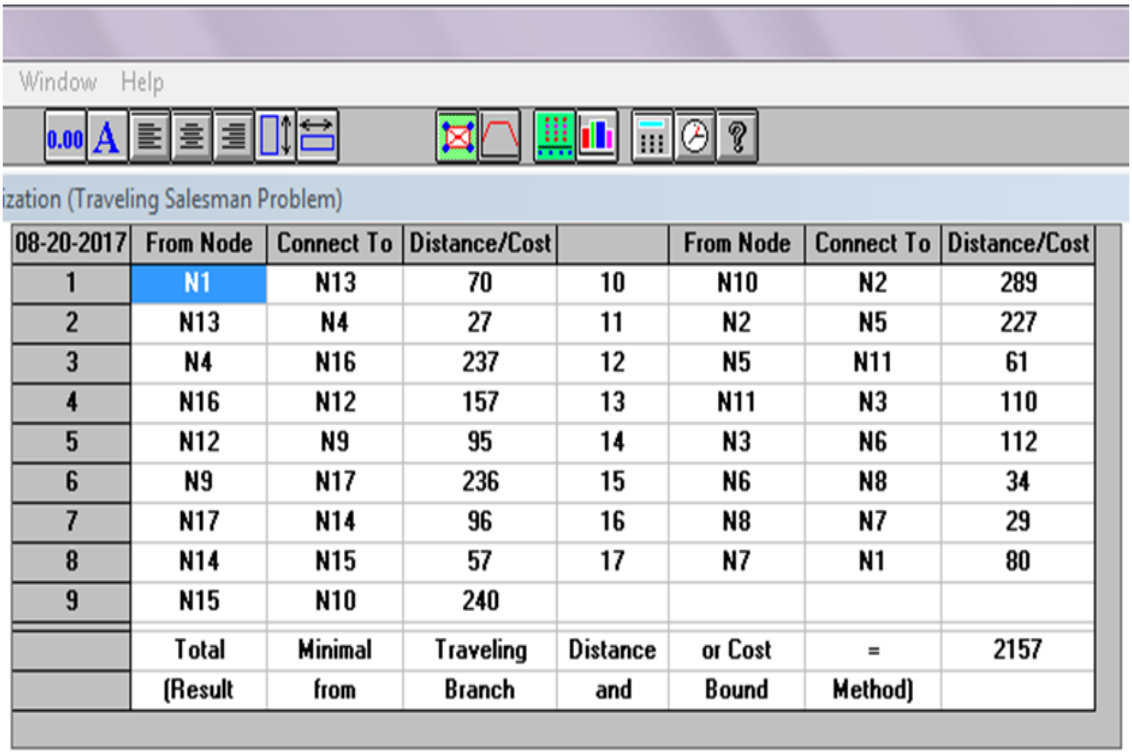


Figura 32: Ventana de WinQSB

Se puede ver en la ventana más pequeña que el WinQSB facilita cuatro métodos diferentes para resolver el Problema del Agente viajero: Nearest Neighbord Heuristic (Heurístico del Vecino más Cercano), Cheapest Insertion Heuristic (Heurístico de Inserción más Barata), Two-way Exchange Improvement Heuristic (Heurístico de Mejora de Dos Intercambios) y el Branch and Bound Method

(Método de Ramificación y acotamiento). En el caso del primero no es muy bueno, da mejores soluciones el algoritmo del vecino más cercano que se ha diseñado en esta investigación.

El mejor método de este software es el Branch and Bound, y es por esa razón que se ha usado para comparar la eficiencia de los algoritmos heurísticos y metaherísticos diseñados en este trabajo para resolver el TSP. Sin embargo, mientras el WinQSB da una solución de 2157 para el "gr17" (véase la siguiente figura), los algoritmos GRASP y Tabu Search alcanzan el óptimo 2085 según la TSPLIB.


Window Help

ization (Traveling Salesman Problem)

08-20-2017	From Node	Connect To	Distance/Cost		From Node	Connect To	Distance/Cost
1	N1	N13	70	10	N10	N2	289
2	N13	N4	27	11	N2	N5	227
3	N4	N16	237	12	N5	N11	61
4	N16	N12	157	13	N11	N3	110
5	N12	N9	95	14	N3	N6	112
6	N9	N17	236	15	N6	N8	34
7	N17	N14	96	16	N8	N7	29
8	N14	N15	57	17	N7	N1	80
9	N15	N10	240				
	Total	Minimal	Traveling	Distance	or Cost	=	2157
	(Result	from	Branch	and	Bound	Method)	

Figura 33: Salida de WinQSB

Siguiendo con el análisis de la tabla, es necesario advertir que aunque el GRASP ha encontrado el mayor número de soluciones óptimas, cuatro en total, no se puede afirmar que sea el método más eficiente. Haría falta estudiar los errores promedio de cada método al resolver las instancias como se muestra en la siguiente tabla:

ERROR PROMEDIO							
	TSPLIB		SOFTWARES		HEURÍSTICOS		METAHEURÍSTICO
NO	INSTANCIA	MEJOR SOLUCIÓN	SOLVER	WINQSB	VECINO	GRASP	TABU
1	brazil58	25395	88.69	15.69	7.83	12.65	7.28
2	dantzig42	699	0.00	13.59	23.61	14.74	22.03
3	fri26	937	1.71	11.63	2.99	0.00	0.00
4	gr17	2085	0.24	3.45	4.46	0.00	0.00
5	gr21	2707	3.55	15.22	10.93	0.00	11.97
6	gr24	1272	9.12	7.86	22.09	6.53	17.14
7	gr48	5046	29.94	15.83	15.74	18.27	14.94
8	gr120	6942	321.51	12.68	21.55	41.69	19.27
9	hk48	11461	46.84	13.56	5.90	20.73	4.72
10	si175	21407	23.14	100.00	2.77	55.52	2.97
11	swiss42	1273	46.43	16.34	12.88	12.10	15.16
12	tsp8	34	0.00	0.00	0.00	0.00	0.00
ERROR PROMEDIO			47.60%	18.82%	10.90%	15.18%	9.62%

Óptimo Alcanzado

Figura 34: Error Promedio

Se consideró oportuno incluir la tabla anterior porque en ella podemos apreciar los errores globales para cada método: Solver (47.60%), WinQSB (18%), Vecino Más Cercano (10.90%), GRASP (15.18%) y Tabu Search (9.62%). En definitiva, el Tabu Search es el mejor de todos ellos, con un 90% de calidad, y algunas de las razones se exponen a continuación.

A medida que aumenta la dimensión del problema el complemento Solver de Excel comienza a fallar. Para la instancia “gr120”, por ejemplo, el error de solución respecto a la TSPLIB es de 321.51%. Esa es la debilidad de este software. En la siguiente gráfica se puede apreciar cómo Solver resultó ser el peor método, seguido por el WinQSB.

Figura 35: Error Promedio por Instancia

Con instancias pequeñas el Solver funciona de la siguiente manera: 1) copiar la instancia en Excel y usar los comandos DATOS y Texto en Columnas para obtener la matriz correspondiente, 2) crear bajo la matriz un vector índice y terminarlo copiando la primera componente, 3) usar la función INDICE para capturar las distancias entre las ciudades, 4) aplicar la función SUMA para obtener la solución del vector índice, 5) utilizar el complemento Solver para optimizar la suma obtenida anteriormente.

Así, en la siguiente figura se puede apreciar que mientras el Solver da una solución de 2803 para el "gr21", el GRASP la mejora reduciéndola al óptimo 2707:

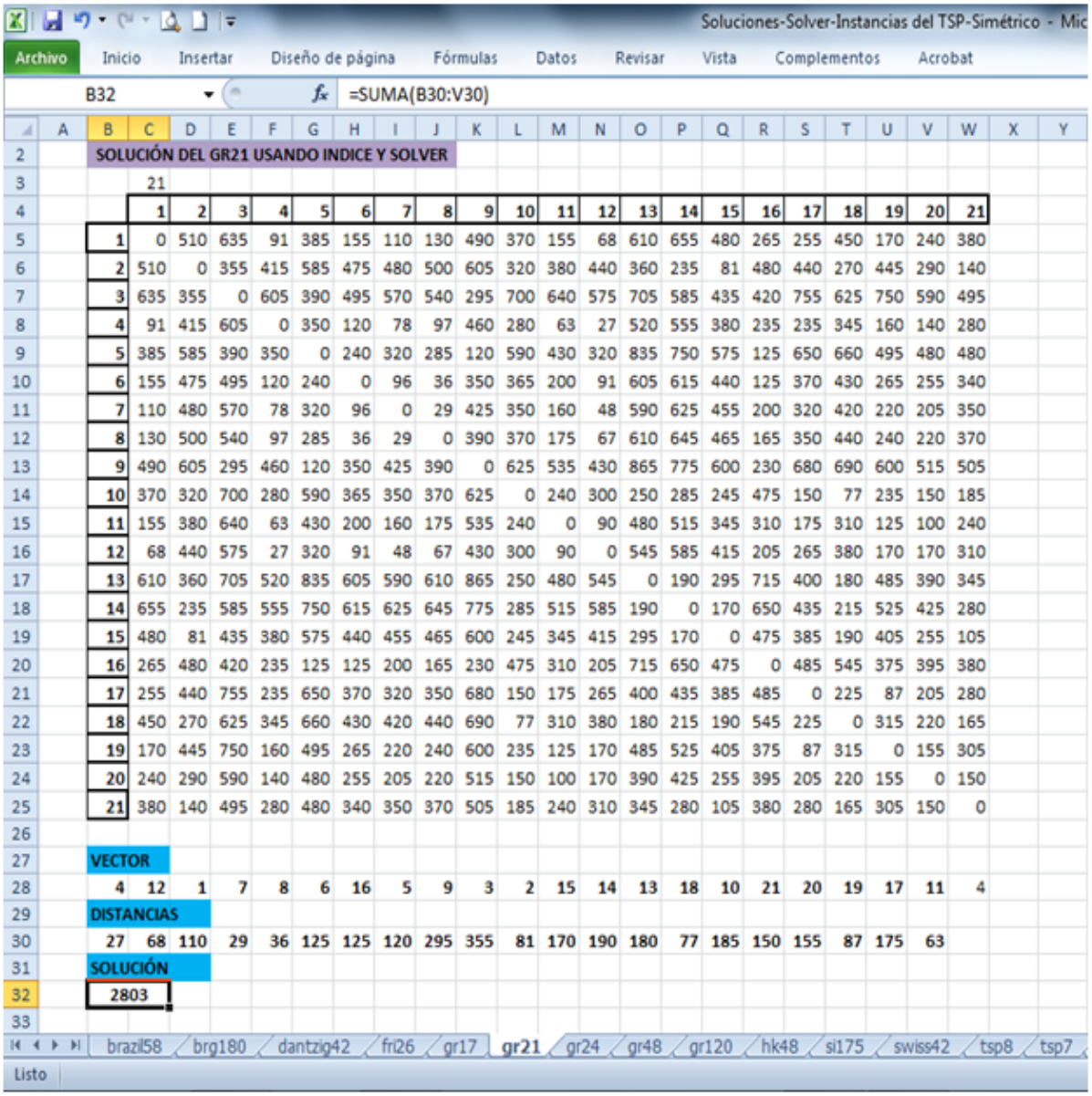


Figura 36: Solución Solver

En cuanto, a los heurísticos y metaheurísticos diseñados, el problema de entrar los datos fue superado porque leen directamente el fichero que contiene la instancia correspondiente. El resto depende de la capacidad de memoria de la computadora que se esté usando. No obstante, las debilidades son otras. En el caso del Vecino Más Cercano se corrió cada instancia con un número máximo de 1000 iteraciones, hacerlo con un número mayor no mejora los resultados. Este algoritmo se queda atrapado en una solución óptima, pero local.

El GRASP, que también es un algoritmo greedy, se queda atrapado eventualmente en un óptimo local. Se requiere de un gran número de iteraciones para mejorar las soluciones, y estas mejoran lentamente. En definitiva, el mejor método que se presenta en esta parte de la investigación es el Tabu Search. Su error promedio al resolver las instancias es de apenas 9.62%, es decir que su efectividad es de 90%, como lo muestra la siguiente gráfica:

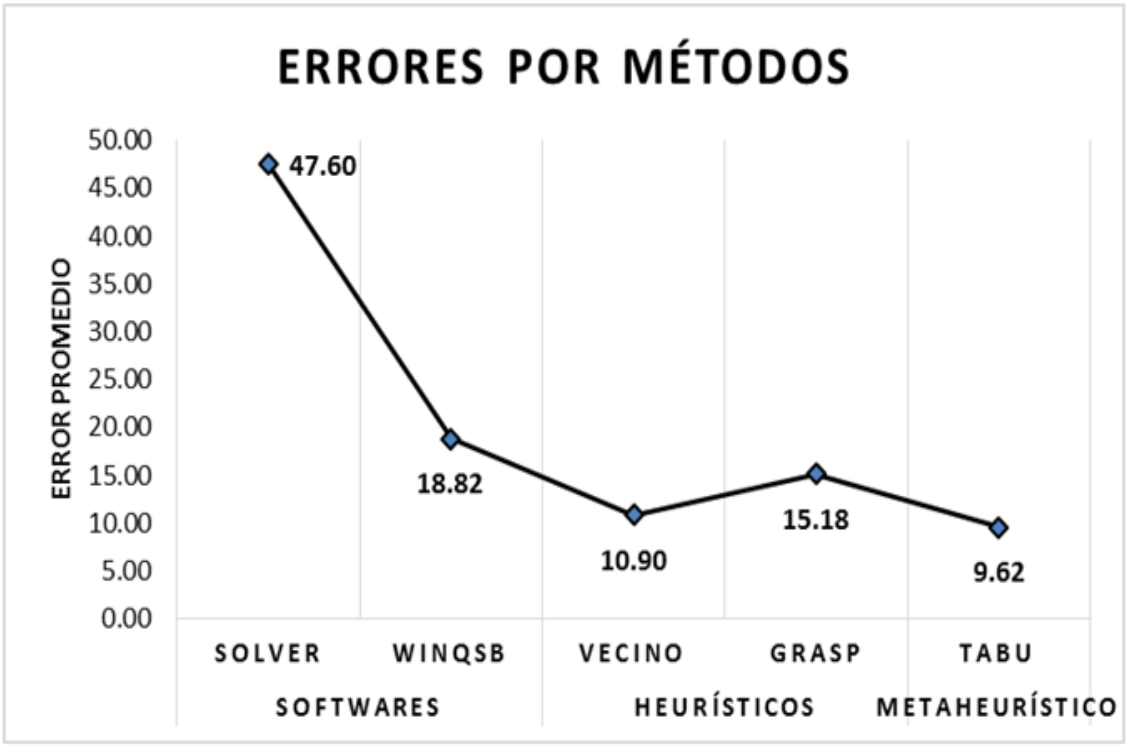


Figura 37: Error promedio por métodos

Y este error podría ser mucho menor, y por ende la eficiencia del algoritmo mucho mayor, en caso de tener una computadora con más memoria y mayor capacidad de procesamiento. El Tabu Search fue ejecutado con solamente el análisis de 200 entornos, con un mayor número genera enormes archivos de escritura que contienen las soluciones. Pero estos archivos no solo contienen las soluciones sino que grandes cantidades de iteraciones por entornos y enormes Lista Tabú, lo que los hace muy pesados e imposibles de abrir para obtener la solución que contienen.

11.2 Comparación del Scatter Search con GRASP y Tabu Search

El scatter Search que se ha utilizado para el TSP tiene las siguientes características especiales:

- La población inicial fue creada a partir de un GRASP, es decir, se consiguió Psize soluciones iniciales distintas derivadas de un método GRASP. Lo que se hizo es efectuar iteraciones Grasp para que nos den Psize soluciones distintas. Recordar que el método Grasp construye una solución a partir de un método determinista al que se le ha añadido un poco de aleatoriedad y después pasando una rutina de mejora. Con lo que la población inicial que se tendrá para el Scatter estará formada por óptimos locales del problema. Quizás este hecho no nos asegure mucha diversidad.
- Cuando se combinan soluciones, para cada solución producida por una combinación, se aplica una rutina de mejora.
- El método utilizado para la combinación de soluciones es un mecanismo de votos.
- La rutina de mejora que se ha utilizado está basada en dos intercambios.

Se compara el Scatter con otros dos metaheurísticos, un GRASP y un Tabu Search. Las pruebas han sido hechas dejando el mismo tiempo para los 3 métodos y haciendo diferentes variaciones de los parámetros de cada uno de los métodos utilizados. Para el Scatter se ha variado solo el tamaño del conjunto de referencia con 10, 20, 30. No se varió el tamaño de Psize ya que al tener que ser óptimos locales en estas instancias determinadas conseguía alrededor de 100 en un tiempo que se puede considerar aceptable, sin embargo para conseguir más tardaba mucho más (por tener que ser óptimos locales diferentes). El GRASP utilizado es del vecino más cercano con un grado de aleatoriedad que va desde 4 (es el menos aleatorio) a 1 (que es el que tiene más aleatoriedad). Se han usado 30 instancias de la TSPLIB de tamaño medio.

Como se observa en la siguiente tabla el algoritmo que obtuvo mejores resultados fue el Scatter Search con parámetros Psize = 100 y RefSet = 20. Además, de tener la media y desviación típica más baja respecto a todos los demás, consigue un mayor número de mínimos valores. Aquí no se habla de valores óptimos porque las instancias tienen las distancias con números decimales y no se pudo leer perfectamente estos valores decimales por tal razón se habla de valor mínimo conseguido con los algoritmos propuestos.

FUNCIÓN OBJETIVO									
	SCATTER			GRASP			TABU		
INSTANCIA	100;10	100;20	100;30	4	2	1	40;1000	80;2000	MÍNIMO
bier127	118288	118256	118336	118613	119016	119116	124831	124831	118256
d198.tsp	15823	15793	15809	15816	15843	15814	16175	16189	15793
eil101.tsp	619	611	611	613	618	625	699	657	611
fl417.tsp	11798	11842	11845	11779	11867	11844	12255	12227	11779
gil262.tsp	2350	2388	2368	2344	2358	2374	2452	2452	2344
kroA100.tsp	21343	21247	21247	21247	21247	21247	22172	22198	21247
kroA150.tsp	26675	26485	26517	26747	26707	26911	28412	28364	26485
kroA200.tsp	29702	29604	29487	29521	29620	29804	31421	31400	29487
kroB100.tsp	22207	22086	22126	22091	22086	22150	23236	23236	22086
kroB150.tsp	26499	26063	26079	26078	26142	26279	29120	28900	26063
kroB200.tsp	30000	29530	29654	29665	30017	30077	30992	30857	29530
kroC100.tsp	20723	20703	20703	20703	20723	20703	23128	23137	20703
kroD100.tsp	21526	21249	21249	21301	21265	21331	22336	22343	21249
kroE100.tsp	22053	22093	22075	22075	22029	22033	22831	22831	22029
lin105.tsp	14392	14354	14354	14354	14354	14354	15062	15062	14354
lin318.tsp	43235	43170	43451	42982	43232	43200	44386	44154	42982
pr107.tsp	44302	44369	44280	44280	44280	44280	45729	45781	44280
pr124.tsp	59011	59011	59011	59011	59011	59011	60232	59882	59011
pr136.tsp	97444	96967	96977	98481	98130	98103	105500	105209	96967
pr144.tsp	58572	58521	58521	58521	58521	58521	59106	59106	58521
pr152.tsp	73780	73759	73622	73622	73622	73626	75196	75196	73622
pr226.tsp	80803	80431	80640	80506	80546	80909	81038	81038	80431
pr264.tsp	49697	49861	49685	49428	49459	50605	53918	53894	49428
pr299.tsp	48580	49376	49366	48892	49418	49990	52686	52306	48580
rat195.tsp	2357	2329	2352	2291	2341	2401	2432	2444	2291
rd100.tsp	7893	7858	7858	7858	7864	7858	8361	8358	7858
rd400.tsp	15883	16037	15971	15555	15768	15968	16362	16320	15555
ts225.tsp	126936	126809	127129	127335	126850	127129	128381	127580	126809
tsp225.tsp	3925	3901	3912	3849	3924	3935	4057	4017	3849
u159.tsp	42499	42045	42045	42045	42118	42289	44362	44355	42045
Media	**37963.8**	**37891.6**	**37909.3**	**37920.1**	**37965.9**	**38082.9**	**39562.3**	**39477.5**	
Desv. Típica	**32750.6**	**32717.5**	**32756.0**	**32884.8**	**32846.2**	**32900.8**	**33906.5**	**33812.2**	

Mejor Valor Encontrado

Figura 38: Función Objetivo Scatter

Se intentó que el tiempo de computación fuera el mismo para todos los algoritmos. Se trabajó asignando un tiempo máximo a cada algoritmo de 2 segundos por cada ciudad de la instancia. Observar que el Scatter puede durar algo más porque si estaba en medio de una iteración se ha permitido que terminara esa iteración. También se tuvo que tener en cuenta el tiempo para generar la población y el tiempo para combinarlas. Ya que al poner un tiempo máximo al Scatter podría darse el caso que le tomara más tiempo intentando encontrar Psize soluciones iniciales diferentes, que combinándolas. Por ello, se ha dejado del tiempo total 1/8 para la generación de la solución inicial y el resto para combinación.

TIEMPO								
	SCATTER			GRASP			TABU	
INSTANCIA	100;10	100;20	100;30	4	2	1	40;1000	80;2000
bier127	18.84	100.07	171.53	254.03	254.03	254.03	25.87	71.13
d198.tsp	109.3	380.2	410.46	396.02	396.07	396.12	101.18	245.41
eil101.tsp	10.71	51.91	67.23	142.86	168.84	202.01	14.11	42.35
fl417.tsp	939.88	1322.34	2681.79	834.97	834.76	835.91	834.26	834.26
gil262.tsp	243.7	555.57	579.52	524.1	524.15	524.1	277.38	524.21
kroA100.tsp	7.53	37.4	78.55	161.15	188.23	200.04	13.85	36.42
kroA150.tsp	25.43	246.89	298.69	300	300.05	300.05	45.2	111.67
kroA200.tsp	88.87	390.35	398.65	400.08	400.03	400.08	99.58	240.85
kroB100.tsp	10.11	61.02	84.31	155.77	182.4	200.03	14.94	43.06
kroB150.tsp	28.61	180.81	299.51	300	300.01	300.06	41.96	109.85
kroB200.tsp	99.63	341.03	384.97	400.02	400.07	400.02	102.49	301.76
kroC100.tsp	6.53	44.11	65.42	154.94	184.66	200.04	14.72	37.4
kroD100.tsp	6.65	32.02	114.57	200.04	179.49	200.03	14.22	37.85
kroE100.tsp	5.72	30.1	93.49	150.88	176.8	200.04	14.12	37.74
lin105.tsp	5.55	36.8	79.26	178.45	207.95	210.04	15.21	42.01
lin318.tsp	647.62	898.53	680.75	636.2	636.54	636.2	473.41	636.37
pr107.tsp	11.2	25.32	71.13	187.29	214.05	214.04	16.97	44.66
pr124.tsp	19.11	51.69	99.52	248.05	248.04	248.04	23.4	66.74
pr136.tsp	20.1	92.49	248.15	272.05	272.04	272.04	33.01	86.12
pr144.tsp	16.26	79.86	157.97	288.03	288.03	288.08	50.31	126.44
pr152.tsp	25.49	124.51	290.29	304.01	304.01	304.06	39.76	108.97
pr226.tsp	122.92	388.7	456.87	452.09	452.2	452.15	140.45	402.22
pr264.tsp	261.17	527.57	728.31	528.05	528.05	528.11	234.59	528.16
pr299.tsp	579.08	749.95	1102.03	598.14	598.3	598.19	377.99	598.24
rat195.tsp	112.76	371.08	395.02	390.09	390.03	390.03	83.04	220.58
rd100.tsp	13.29	34.55	64.15	157.25	181.97	200.03	13.79	39
rd400.tsp	886.77	959.77	1952.65	800.43	800.48	800.27	800.15	800.16
ts225.tsp	114.46	442.92	515.03	450	450.06	450.11	158.35	359.22
tsp225.tsp	134.41	431.17	436.33	450	450.01	450.17	146.21	326.97
u159.tsp	35.15	167.97	295	318.02	318.02	318.01	44.44	119.85
Media	**153.6**	**305.2**	**443.4**	**354.4**	**361.0**	**365.7**	**142.2**	**239.3**
Desv. Típica	**254.2**	**321.2**	**562.2**	**185.2**	**178.6**	**174.0**	**211.9**	**237.2**

Mejor Valor Encontrado

Figura 39: Tiempo de Computación Scatter

Se consideró oportuno incluir las evaluaciones que cada uno de los algoritmos efectúa de la función objetivo. Existe una gran diferencia entre las que hace por ejemplo el Scatter de parámetro 4 (el que obtiene mejores valores) y el Scatter Search sobre todo el parámetro 100, 30. El Grasp de parámetro 4 produce tan pocas evaluaciones de la función objetivo porque al ser muy determinista se queda bastante cerca de un óptimo local con lo que la fase de mejora de cada solución será muy pequeña. Sin embargo, si se tiene una solución producida por la combinación de dos soluciones puede quedar muy alejada del óptimo por tanto se necesitan muchas iteraciones de la fase de mejora para llegar a un óptimo local.

EVALUACIONES								
	SCATTER			GRASP			TABU	
INSTANCIA	100;10	100;20	100;30	4	2	1	40;1000	80;2000
bier127	4159786	25202082	46256497	8107530	12000912	20266701	8600592	17059416
d198.tsp	21383715	79407026	103455188	7870217	12214071	19510600	22627804	41992725
eil101.tsp	2748140	10786390	19662406	5204273	8954532	19953187	5188648	10778327
fl417.tsp	63013423	84795392	193166711	7457737	10329938	16239697	63100651	58611959
gil262.tsp	28789405	71712193	76274585	5825749	8148147	13404085	39326029	67726276
kroA100.tsp	2196620	13656780	28506569	6570354	10249953	16277626	5196492	9970860
kroA150.tsp	4428958	58469541	83512645	7237522	10019810	16050868	13265481	23752158
kroA200.tsp	13164281	77336420	84059363	7020721	9560189	15425880	21751008	41847048
kroB100.tsp	3504424	21911845	31070966	6231545	10143085	16753736	5623468	11770950
kroB150.tsp	6507306	49479084	85395894	8167793	10728454	16512049	12140727	23255943
kroB200.tsp	19686931	65257265	82890702	6910522	9547133	15458715	21613094	49412616
kroC100.tsp	1714091	16840266	22295171	5748641	9870779	16461055	5409980	9922340
kroD100.tsp	1754193	10855497	40528773	5232435	9325567	16905255	5249864	9941748
kroE100.tsp	1321134	9873561	35798848	5544912	9103215	12111491	5080044	9932044
lin105.tsp	1309051	13364794	27027475	6134801	9890259	16158279	5608779	10965779
lin318.tsp	62469153	92422346	65730786	6058042	8289276	13548586	56999006	66465449
pr107.tsp	3249053	8742482	24685262	6814084	12853369	18166588	6155996	11794354
pr124.tsp	5410086	17023354	34038067	5031146	7839314	15485252	7976752	15840944
pr136.tsp	4384325	21736797	80847879	8509542	11650932	20332394	10032014	19430808
pr144.tsp	3029637	23000355	44316373	5387954	8430987	13640100	10935858	21089858
pr152.tsp	5655398	31481341	78163454	7719455	11315906	16768426	12435948	23988468
pr226.tsp	20741266	67143619	88886458	6171556	10328649	17852082	28451929	53652929
pr264.tsp	33739047	73589222	98045414	5628046	7874839	17598975	40759082	71664320
pr299.tsp	62362003	87744025	130732724	6092539	7844315	15273983	50316798	65982714
rat195.tsp	20267586	89306000	95326821	5545243	8777996	17634915	21118416	40570574
rd100.tsp	4092527	11638274	22820222	6152133	9872607	16952201	5167380	10523988
rd400.tsp	56924458	62135990	136914178	4490265	6154520	10457214	67253494	59075088
ts225.tsp	16763132	83493846	94096338	3125152	5726387	15662185	29972400	56872629
tsp225.tsp	19292680	78946006	79239160	7175712	10346408	17826168	28124102	53875389
u159.tsp	7401482	45051648	82896792	6175659	9851679	17690106	13669208	26829852
Media	**16715443**	**46746781**	**70554724**	**6311376**	**9574774**	**16412613**	**20971702**	**331532512**
Desv. Típica	**19480702**	**30006155**	**39503648**	**1165829**	**1602529**	**2225421**	**18241772**	**21484643**

Mejor Valor Encontrado

Figura 40: Evaluación

En definitiva, tomando en cuenta los resultados de las tres tablas anteriores (Función Objetivo, Tiempo de Computación y Evaluaciones), podemos afirmar categóricamente que el Scatter Search es un algoritmo más eficiente que el GRASP y el Tabu Search.

12 CONCLUSIONES

- Se realizó una revisión exhasutiva del estado del arte para el Problema del Agente Viajero (TSP). Este es uno de los problemas más importantes de la optimización combinatoria, tanto por su complejidad computacional, como por sus múltiples aplicaciones. Actualmente hay disponibles en las revistas indexadas una cantidad enorme de propuestas científicas para resolver el TSP, desde nuevos teoremas sobre su estructuta, asi como aproximaciones muy buenas como planos de corte, Simulated annealing, GRASP, Tabu Search, Scatter Search, etc. Dentro de los científicos que más tiempo y esfuerzo le han dedicado a este problema están Fred Glover y Manuel Laguna de la Universidad de Colorado (EEUU), asi como Rafael Martí de la Universidad de Valencia (España).

- Se estructuraron los resultados teóricos de Análisis Convexo para formular y resolver el Problema del Agente Viajero. Dicha estructura era realmente necesaria dado que el TSP se puede plantear como un problema de grafos, y estos a su vez tienen formas convexas.

- Se diseñaron los algoritmos Heurísticos del Vecino Más Cercano y Greedy Randomized Adaptive Search Procedures (GRASP), y los Metaheurísticos Tabu Search y Scatter Search para resolver el TSP. Estos funcionaron muy bien sobre las instancias de la librería TSPLIB, alcanzando algunos de los óptimos y en promedio, el algoritmo Tabu Search que es el más prometedor para instancias de tamaño pequeño y mediano alcanza un índice del 90% de calidad. En el caso de este algoritmo la fase de mejora funciona utilizando ideas de inteligencia artificial, en cuanto a la memoria a corto plazo que implementa mediante estructuras simples con el objetivo de dirigir la búsqueda teniendo en cuenta la historia o registro computacional de ésta, a través de la Lista Tabú evitando quedarse atrapado en un óptimo local. Con respecto al Scatter Search, es el mejor para instancias grandes, tiene el mismo orden de eficiencia que el Tabu Search. Realiza la búsqueda de mejores soluciones mediante exploración sistemática sobre una serie de buenas soluciones llamadas conjunto de referencia que se ha de ir actualizando.

- Se comparó la calidad de los algoritmos diseñados para el TSP, al implementar y obtener resultados usando lenguaje C++ (ambiente de programación Microsoft Visual Studio 2008), Solver de Excel y WinQSB, utilizando las instancias de Traveling Salesman Problem Library (TSPLIB). Se hace notar que las computadoras disponible para esta investigación no eran ni de las más modernas ni las más rápidas, de haber contado con hardware disponibles como en universidades desarrolladas y de primer mundo se habría llegado al óptimo de la mayoría de las instancia, en un tiempo razonable.

- El análisis de los resultados obtenido con los algoritmos heurísticos y metaheurísticos, indica que el Vecino más Cercano, GRASP y Tabu Search, son más eficientes que los optimizadores

WinQSB y Solver de Excel. No obstante, el mejor algoritmo diseñado para instancias pequeñas y medianas fue el Tabu Search y para instancias grandes fue el Scatter Search.

- Tomando en cuenta la eficiencia que debe tener un método para resolver el Problema del Agente Viajero, se concluye que el orden decreciente de eficiencia es el siguiente: Scatter Search, Tabu Search, GRASP, Vecino más Cercano, WinQSB y Solver.

13 PERSPECTIVAS DE FUTURO SOBRE EL TSP

- Para investigaciones futuras se planea trabajar en las Fases de Mejora de los algoritmos diseñados, incorporando nuevas ideas de inteligencia artificial. También se harán las adecuaciones necesarias para tratar con las instancias asimétricas de la TSPLIB.

14 BIBLIOGRAFÍA

Álvarez, R., Corberán, A., Tamarit, J. (1985). La combinatoria poliédrica y el problema del viajante. Aplicación al caso de ciento tres ciudades Españolas. Qüestió, v. 9, n. 3, p. 199-213, 1985.

Alvarez-Valdés, R., & Parajón, A. (2002). A tabu search algorithm for large-scale guillotine (un) constrained two-dimensional cutting problems. Computers & Operations Research, 29(7), 925-947.

Alvarez-Valdes, R., Martí, R., Tamarit, J. M., & Parajón, A. (2007). GRASP and path relinking for the two-dimensional two-stage cutting-stock problem. INFORMS Journal on Computing, 19(2), 261-272.

Alvarez-Valdes, R., Parajon, A., & Tamarit, J. M. (2001, July). A computational study of heuristic algorithms for two-dimensional cutting stock problems. In 4th metaheuristics international conference (MIC'2001) (pp. 16-20).

Alvarez-Valdes, R., Parajon, A., & Tamarit, J. M. (2002). A computational study of LP-based heuristic algorithms for two-dimensional guillotine cutting stock problems. OR Spectrum, 24(2), 179-192.

Applegate, D. Bixby, R. Chvátal, V. Cook. (2006). The Traveling Salesman Problem.

Avila, J. (1995). Método de Karmarkar Revista de Matemática: Teoría y Aplicaciones 2(1): 45-55.

Bazaraa, M.S., Jarvis, J.J., Sherali, H. (1990). Linear Programming and Network Flows, 2nd Edition. John Wiley & Sons, New York.

Bertazzi, L., Paletta, G., Speranza, M.G. (2004). An improved heuristic for the period traveling salesman problem. Computers & Operations Research. Vol. 31, 2004, p.1215-1222.

Calviño, A. (2011). Cooperación en los problemas del viajante (TSP) y de rutas de vehículos (VRP): una panorámica. Santiago de Compostela.

Cassini, L., Righini, G. (2004). Heuristic algorithms for the TSP with rear-loading. In: ANNUAL CONFERENCE OF THE ITALIAN OPERATIONAL RESEARCH SOCIETY (AIRO XXXV), 35, 2004, Lecce, Italy.

Chao, I.M., Golden, B.L., Wasil, E.A. (1995). A new heuristic for the period traveling salesman problem. Computers & Operations Research. Vol.22, 1995b, p.553-565.

Chen S., Chien, C. Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. Expert Systems with Applications, v. 38, n. 12, p. 14439-14450, 2011. http://dx.doi.org/10.1016/j. eswa.2011.04.163

Cook S.A. (1971). The complexity of theorem-proving procedures. ACM Press, New York, USA. 151-158.

Dantzig, G., Fulkerson, R., Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. Journal of the operations research society of America, v. 2, n, 4, p. 393 410.

Feo, T. and Resende, M.G.C. (1989), A probabilistic heuristic for a computational difficult set covering problems, Operations research letters, 8, 67–71.

Feo, T. and Resende, M.G.C. (1995), Greedy Randomized Adaptive Search Procedures, Journal of Global Optimization, 2, 1–27.

Glover, F. (1986) "Future Paths for Integer Programming and Links to Artificial Intelligence", Computers and Operations Research. 5. 533-549.

Glover, F. and Laguna, M. (1997). Tabu Search, Kluwer Academic Publishers.

Glover, F., Laguna, M. and Martí R. (2006) Principles of Tabu Search. To appear in Approximation Algorithms and Metaheuristics, Chapman & Hall/CRC.

González, C., Sánchez, M. (1991). El método de Karmarkar: un estudio de sus variantes.

Guevara, R. A. P. (2001). Algoritmos heurísticos eficientes para problemas de corte en dos dimensiones (Doctoral dissertation, Universitat de València).

Held, M., Karp, R.M. (1962). A dynamic programming approach to sequencing problems. Journal of the Society of Industrial an Applied Mathematics 10, 196-210.

Jimenez, G. (2009). Optimización. Universidad Nacional de Colombia.

Martí, R. (2003). Procedimientos metaheurísticos en optimización combinatoria. Matemátiques, v. 1, n. 1, p. 1-60.

Osman, I.H. and Kelly, J.P. (eds.) (1996), Meta-Heuristics: Theory and Applications, Ed. Kluwer Academic, Boston.

Parajón Guevara, A. (2009). El Álgebra y su Tratamiento Metodológico y sus Aplicaciones. Módulo III. Managua, Nicaragua.

Parreño Torres, F. (2004). Algoritmos heurísticos y exactos para problemas de corte no guillotina en dos dimensiones.

Pérez, J. (2011). Heurística inspirada en el análisis sistémico del "Vecino más cercano", para solucionar instancias simétricas TSP, empleando una base comparativa multicriterio. Universidad Nacional de Colombia, Medellín.

Pérez, J., Jaramillo, G. (2011). Espacio literario relevante sobre el problema del vendedor viajero (TSP): contenido, clasificación, métodos y campos de inspiración.

Reeves, C.R. (1995), Modern Heuristic Techniques for Combinatorial Problems, Ed. McGraw-Hill, UK.

Ríos, R., González, J. (2000). Investigación de operaciones en acción: Heurísticas para la solución del TSP.

Rosenkrantz, D. J., Stearns, R. E., & Lewis, II, P. M. (1977). An analysis of several heuristics for the traveling salesman problem. SIAM journal on computing, 6(3), 563-581. Taha, H. (2011). Investigación de operaciones. University of Arkansas, Fayetteville. Pearson. Novena edición.

Zanakins, S. H. y Evans, J. R. (1981). Heuristic Optimization: Why, When and How to Use It. Interfaces. Vol. 11(5) .

15 ANEXOS

15.1 Código C++ para el Algoritmo del Vecino Más Cercano

```
// 01-Algoritmo del Vecino Más Cercano.cpp: define el punto de
entrada de la aplicación de consola. //

// 01-INICIO DE LIBRERIAS //
#include "stdafx.h"
#include <malloc.h>
#include <stdio.h>
#include <tchar.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#define getrandom(min,max)((rand()%(int)(((max)+1)-(min)))+(min))
// 01-FIN DE LIBRERIAS //

// 02-INICIO DE CABECERAS DE FUNCIONES //
void ProblemaMemoria(char *texto);
void LiberaMatriz(int **matriz, int dim);
int *ReservaVector(int dim);
void EscribeMatriz(FILE *pescribir,int dim1,int dim2,int **matriz);
int EstaRepite(int TamañoVectorSolucion,int SiguienteNodo,int *v);
void EscribePermutacion(FILE *pescribir, int dim, int *w);
void CopiaPermutacion(int dim,int *v1, int *v2);
void SolucionVecinoMasCercano(int **m, int *v, int dim);
void ValorSolucion(FILE *pescribir,int **m,int dim,int *v,int *valor);
// 02-FIN DE CABECERAS DE FUNCIONES //

// 03-INICIO DE MAIN //
void main(int argc,char **argv)
{
    // 03.1-DECLARACION DE VARIABLES //
    int dim,**m,*x,*cx,fx;
    int i,j,valor;
    FILE *pescribir;
    FILE *pleer;
```

```
time_t tstart1,cstart1,tend1,cend1;

// 03.2-ABRIR ARCHIVO DE ESCRITURA //
pescribir=fopen("01-VECINO.txt","w");
if(pescribir==NULL)
    ProblemaMemoria("PROBLEMAS AL RESERVAR MEMORIA");

// 03.3-ABRIR ARCHIVO DE LECTURA //
pleer=fopen("gr17.txt","r");
if(pescribir==NULL)
    ProblemaMemoria("PROBLEMAS AL RESERVAR MEMORIA");
fscanf(pleer,"%d",&dim);
fprintf(pescribir,"SOLUCIÓN DEL TSP-%d", dim);
fprintf(pescribir," USANDO EL VECINO MÁS CERCANO \n\n");
fprintf(pescribir,"LA DIMENSIÓN ES %d\n\n", dim);

// 03.4-RESERVA MATRIZ DINÁMICA //
m=(int**)calloc(dim,sizeof(int*));
if(m==NULL)
    printf("\n NO HAY ESPACIO  PARA RESERVAR MATRIZ");

// 03.5-LEE MATRIZ //
for(i=0;i<dim;i++)
    m[i]=ReservaVector(dim);
    for(i=0;i<dim;i++)
    {
        fscanf(pleer,"\n\n");
        for(j=0;j<dim;j++)
            fscanf(pleer,"%d",&m[i][j]);
    }
fclose(pleer);

// 03.6-ESCRIBE MATRIZ //
EscribeMatriz(pescribir,dim,dim,m);
cx=ReservaVector(dim);
fprintf(pescribir,"\n\n");
```

```
// 03.7-COMIENZA EL RELOJ //
time(&tstart1);
cstart1=clock();

// 03.8-SOLUCIÓN CON EL VECTOR ÍNDICE //
x=ReservaVector(dim);
for(i=0;i<dim;i++)
    x[i]=i+1;
fprintf(pescribir,"EL VECTOR ÍNDICE Y SU VALOR SOLUCIÓN ES: \n\n");
ValorSolucion(pescribir,m,dim,x,&fx);
fprintf(pescribir,"\n\n");

// 03.9-VECINO MÁS CERCANO //
fprintf(pescribir,"UN VECTOR CON EL VECINO MÁS CERCANO Y SU
SOLUCIÓN ES: \n\n");
SolucionVecinoMasCercano(m,x,dim);
ValorSolucion(pescribir,m,dim,x,&fx);
fprintf(pescribir,"\n\n");

// 03.10-FASE CONSTRUCTIVA: ITERACIONES CON EL VECINO MÁS CERCANO //
fprintf(pescribir,"VECTORES GENERADOS CON EL VECINO MÁS
CERCANO SON: \n\n");
int MejorSol=1000000, numit=1000;
for(i=0;i<numit;i++)
{
    SolucionVecinoMasCercano(m,x,dim);
    ValorSolucion(pescribir,m,dim,x,&fx);
    if(fx<MejorSol)
    {
        MejorSol=fx;
        CopiaPermutacion(dim,x,cx);
    }
}
fprintf(pescribir,"\n\n");
fprintf(pescribir,"LA MEJOR SOLUCIÓN DESPUÉS DE %d", numit);
fprintf(pescribir," ITERACIONES ES: \n\n");
ValorSolucion(pescribir,m,dim,cx,&MejorSol);
```

```
    fprintf(pescribir,"\n\n");

    // 03.11-TERMINA EL RELOJ //
    time(&tend1);
    cend1=clock();
    fprintf(pescribir,"\ntiempo = %4.23fseg/",
    ((float)cend1-cstart1)/CLK_TCK);

    // 03.12-LIBERA MEMORIA //
    free(x);
    free(cx);
    LiberaMatriz(m,dim);

    // 03.13-CIERRA LOS ARCHIVOS DE SALIDA //
    fclose(pescribir);
}
// 03-FIN DE MAIN //

// 04-INICIO DE LAS FUNCIONES AUXILIARES //

// 04.1-PROBLEMAS AL RESERVAR MEMORIA //
void ProblemaMemoria(char *texto)
{
    printf("\n%s",texto);
    exit(0);
}

// 04.2-LIBERA MATRIZ //
void LiberaMatriz(int **matriz, int dim)
{
    int i;
    for(i=0;i<dim;i++)
    {
        free(matriz[i]);
    }
    free(matriz);
}
```

```
// 04.3-RESERVA MEMORIA PARA EL VECTOR //
int *ReservaVector(int dim)
{
    int *v;
    v=(int*)calloc(dim,sizeof(int));
    if(v==NULL)
        printf("problemas en lectura");
    return(v);
}

// 04.4-ESCRIBE MATRIZ //
void EscribeMatriz(FILE *pescribir, int dim1,int dim2,int **matriz)
{
    int i,j;
    fprintf(pescribir,"MATRIZ DE %d FILAS Y %d COLUMNAS:
    \n\n",dim1,dim2);
    for(i=0;i<dim1;i++)
    {
        for(j=0;j<dim2;j++)
        {
            fprintf(pescribir,"%6d",matriz[i][j]);
        }
        fprintf(pescribir,"\n\n");
    }
}

// 04.5-BUSCA SI EL NUEVO NODO SE REPITE O NO //
int EstaRepite(int TamañoVectorSolucion,int SiguienteNodo,int *v)
{
    int i;
    for(i=0;i<TamañoVectorSolucion;i++)
    {
        if(v[i]==SiguienteNodo)
        return(1);
    }
    return(0);
```

```
}

// 04.6-ESCRIBE PERMUTACIÓN //
void EscribePermutacion(FILE *pescribir, int dim, int *w)
{
    int i;
    fprintf(pescribir,"\n\n");
    for(i=0;i<dim;i++)
    {
        fprintf(pescribir,"%4d",w[i]);
    }
    fprintf(pescribir,"\n\n");
}

// 04.7-COPIA LA PERMUTACIÓN X //
void CopiaPermutacion(int dim, int *v1, int *v2)
{
    int i;
    for(i=0;i<dim;i++)
    {
        v2[i]=v1[i];
    }
}

// 04.8-ENCUENTRA EL NUEVO NODO MÁS CERCANO //
void SolucionVecinoMasCercano(int **m, int *v, int dim)
{
    int nodoin,er,r,k,ind;
    long int min;
    nodoin=getrandom(1,dim);
    v[0]=nodoin;
    for(r=1;r<dim;r++)
    {
        min=100000;
        for(k=0;k<dim;k++)
        {
            er=EstaRepite(r,k+1,v);
```

```
            if(er==0 && m[nodoin-1][k]!=0 && m[nodoin-1][k]<min)
            {
                min=m[nodoin-1][k];
                ind=k;
            }
        }
        nodoin=ind+1;
        v[r]=nodoin;
    }
}

// 04.9-CALCULA EL VALOR DE LA SOLUCIÓN //
void ValorSolucion(FILE *pescribir, int **m, int dim, int *v, int *valor)
{
    int i;
    *valor=m[v[0]-1][v[dim-1]-1];
    for(i=0;i<dim-1;i++)
    {
        *valor=*valor+m[v[i]-1][v[i+1]-1];
    }
    for(i=0;i<dim;i++)
    {
        fprintf(pescribir,"%6d",v[i]);
    }
    fprintf(pescribir,"   LA SOLUCIÓN ES: %6d \n",*valor);
}
// 04-FIN DE LAS FUNCIONES AUXILIARES //
```

15.2 Código C++ para el Algoritmo del GRASP

```
// 02-Algoritmo del GRASP.cpp: define el punto de entrada de la
 aplicación de consola. //

// 01-INICIO DE LIBRERIAS //
#include "stdafx.h"
#include <malloc.h>
#include <stdio.h>
#include <tchar.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
# define getrandom(min,max)((rand()%(int)(((max)+1)-(min)))+(min))
// 01-FIN DE LIBRERIAS //

// 02-INICIO DE CABECERAS DE FUNCIONES //
void ProblemaMemoria(char *texto);
void LiberaMatriz(int **matriz, int dim);
int *ReservaVector(int dim);
void EscribeMatriz(FILE *pescribir,int dim1,int dim2,int **matriz);
int *fun_reserva_vector(int dim);
void fun_escribe_matriz_dis(FILE *ps,int dim1,int dim2,int **matriz);
int fun_numero_aleatorio(int m,int n);
int fun_esta_en_tour(int size,int verso,int *vs);
void fun_ver_candidatos(FILE *ps,int ciu,int verso ,int *vc);
int fun_verdime(FILE *ps,int ciu,int verso,int **matriz,int I,
int *vs);
int fun_nuevo_ver_so(FILE *ps,int ciu,int verso,int **matriz,
int I,int *vs,int *vcr,int dime);
void fun_escribe_solucion(FILE *ps,int ciu,int *vs,int **matriz);
int fun_distancia(FILE *ps,int ciu,int *vs,int **matriz);
void  funcion_copia_vector(FILE *ps,int ciu,int *vs,int *vss);
void funcion_intercambio(FILE *ps,int inicio,int fin,int *vs);
void funcion_reorden(FILE *ps,int inicio,int fin,int *vs);
void escribe_vector(FILE *ps,int ciu,int *vs);
// 02-FIN DE CABECERAS DE FUNCIONES //
```

```
// 03-INICIO DE MAIN //
void main(int argc,char **argv)
{
    // 03.1-DECLARACIÓN DE VARIABLES //
    int ciudades,**matriz,ver_az,ver_so,*vso,*vca,*vcar,i,di_me;
    int coste,costea,a,b,c,d,j,antiguo,nuevo,*vsoa,*vsoaa;
    int k,solucion_grasp,*vsgrasp;
    FILE *ps;
    FILE *pleer;
    time_t tstart1,cstart1,tend1,cend1;

    // 03.2-ABRIR ARCHIVO DE ESCRITURA //
    ps=fopen("02-GRASP.txt","w");
    if(ps==NULL)
        ProblemaMemoria("PROBLEMAS AL RESERVAR MEMORIA");

    // 03.3-ABRIR ARCHIVO DE LECTURA //
    pleer=fopen("gr17.txt","r");
    if(ps==NULL)
        ProblemaMemoria("PROBLEMAS AL RESERVAR MEMORIA");
    fscanf(pleer,"%d",&ciudades);
    fprintf(ps,"LA DIMENSIÓN ES %d\n\n", ciudades);

    // 03.4-RESERVA MATRIZ DINÁMICA //
    matriz=(int**)calloc(ciudades,sizeof(int*));
    if(matriz==NULL)
        printf("\n NO HAY ESPACIO  PARA RESERVAR MATRIZ");

    // 03.5-LEE MATRIZ //
    for(i=0;i<ciudades;i++)
        matriz[i]=ReservaVector(ciudades);
        for(i=0;i<ciudades;i++)
        {
            fscanf(pleer,"\n\n");
            for(j=0;j<ciudades;j++)
                fscanf(pleer,"%d",&matriz[i][j]);
```

```
    }
    fclose(pleer);
EscribeMatriz(ps,ciudades,ciudades,matriz);
fprintf(ps,"\n\n");

// 03.6-LEE MATRIZ //
vso=fun_reserva_vector(ciudades);
vca=fun_reserva_vector(ciudades-1);
vcar=fun_reserva_vector(ciudades);
vsoa=fun_reserva_vector(ciudades);
vsoaa=fun_reserva_vector(ciudades);
vsgrasp=fun_reserva_vector(ciudades);

// 03.7-ITERACIONES DEL GRASP //
time(&tstart1);
cstart1=clock();
solucion_grasp=1000000;
for(k=1;k<1000;k++)
{
    // FASE CONSTRUCTIVA DEL GRASP //
    ver_az=fun_numero_aleatorio(1,ciudades);
    ver_so=ver_az;
    fun_ver_candidatos(ps,ciudades,ver_so,vca);
    for(i=0;i<ciudades-1;i++)
    {
        vso[i]=ver_so;
        di_me=fun_verdime(ps,ciudades,ver_so,matriz,i,vso);
        ver_so=fun_nuevo_ver_so(ps,ciudades,ver_so,matriz,i,
        vso,vcar,di_me);
        vso[i+1]=ver_so;
    }
    fprintf(ps,"\n\n*** Fase Constructiva %d ***", k);
    fun_escribe_solucion(ps,ciudades,vso,matriz);

    // FASE DE MEJORA DEL GRASP //
    costea=1000000;
    for(i=0;i<ciudades-2;i++)
```

```
{
    a=vso[i];
    b=vso[i+1];
    for(j=i+2;j<ciudades;j++)
    {
        c=vso[j];
        if(j+1<ciudades) d=vso[j+1];
        else d=vso[0];
        antiguo=matriz[a-1][b-1]+matriz[c-1][d-1];
        nuevo=matriz[a-1][c-1]+matriz[b-1][d-1];
        if(nuevo<antiguo)
        {
            coste=fun_distancia(ps,ciudades,vso,matriz)
            -antiguo+nuevo;
            if(coste<costea)
            {
                costea=coste;
                funcion_copia_vector(ps,ciudades,vso,vsoa);
                funcion_intercambio(ps,i+1,j,vsoa);
                if(j-(i+1)<=2)
                    funcion_copia_vector(ps,ciudades,vsoa,vsoaa);
                else
                {
                    funcion_reorden(ps,i+2,j,vsoa);
                    funcion_copia_vector(ps,ciudades,vsoa,vsoaa);
                }
            }
        }
    }
}
fprintf(ps,"\n*** Fase de Mejora  %d ***",k);
fun_escribe_solucion(ps,ciudades,vsoaa,matriz);
if(fun_distancia(ps,ciudades,vsoaa,matriz)<solucion_grasp)
{
    solucion_grasp=fun_distancia(ps,ciudades,vsoaa,matriz);
    funcion_copia_vector(ps,ciudades,vsoaa,vsgrasp);
}
```

```
    }
    fprintf(ps,"\n\n\n*** Después de %d Iteraciones GRASP ***",k);
    fun_escribe_solucion(ps,ciudades,vsgrasp,matriz);
    time(&tend1);
    cend1=clock();
    fprintf(ps,"\ntiempo = %4.2fseg/",((float)cend1-cstart1)/CLK_TCK);

    // 03.8-LIBERA MEMORIA //
    LiberaMatriz(matriz,ciudades);

    // 03.9-CIERRA LOS ARCHIVOS DE SALIDA DE ESCRITURA//
    fclose(ps);
}
// 03-FIN DE MAIN //

// 04-INICIO DE LAS FUNCIONES AUXILIARES //

// 04.1-PROBLEMAS AL RESERVAR MEMORIA //
void ProblemaMemoria(char *texto)
{
    printf("\n%s",texto);
    exit(0);
}

// 04.2-LIBERA MATRIZ //
void LiberaMatriz(int **matriz, int dim)
{
    int i;
    for(i=0;i<dim;i++)
    {
        free(matriz[i]);
    }
    free(matriz);
}

// 04.3-RESERVA MEMORIA PARA EL VECTOR //
int *ReservaVector(int dim)
```

```
{
    int *v;
    v=(int*)calloc(dim,sizeof(int));
    if(v==NULL)
        printf("Problemas en Lectura");
    return(v);
}

// 04.4-ESCRIBE MATRIZ //
void EscribeMatriz(FILE *ps, int dim1,int dim2,int **matriz)
{
    int i,j;
    fprintf(ps,"MATRIZ DE %d FILAS Y %d COLUMNAS: \n\n",dim1,dim2);
    for(i=0;i<dim1;i++)
    {
        for(j=0;j<dim2;j++)
        {
            fprintf(ps,"%6d",matriz[i][j]);
        }
        fprintf(ps,"\n\n");
    }
}

// 04.5-FUNCIÓN RESERVA VECTOR //
int *fun_reserva_vector(int dimension)
{
    int *reserva_vector;
    reserva_vector=(int*)calloc(dimension,sizeof(int));
    if(reserva_vector==NULL)
        printf("Problemas en Lectura");
    return(reserva_vector);
}

// 04.6-FUNCIÓN ESCRIBE MATRIZ DE DISTANCIAS //
void fun_escribe_matriz_dis(FILE *ps,int dim1,int dim2,int **matriz)
{
    int i,j;
```

```
    fprintf(ps,"Matriz de Distancias Entre %d Ciudades\n\n",dim1);
    for(i=0;i<dim1;i++)
    {
        for(j=0;j<dim2;j++)
        {
            fprintf(ps,"%3d",matriz[i][j]);
        }
        fprintf(ps,"\n");
    }
}

// 04.7-FUNCIONES DE LA FASE CONSTRUCTIVA DEL GRASP //

int fun_numero_aleatorio(int m,int n)
{
    int numero_aleatorio;
    numero_aleatorio=getrandom(m,n);
    return(numero_aleatorio);
}

void fun_ver_candidatos(FILE *ps,int ciu,int verso ,int *vc)
{
    int i,j;
    j=0;
    for(i=0;i<ciu;i++)
    {
        vc[i]=i+1;
        if(vc[i]!=verso)
        {
            vc[j]=i+1;
            j++;
        }
    }
}

int fun_verdime(FILE *ps,int ciu,int verso,int **matriz,int I,int *vs)
{
```

```
    int a,j,verme,d=1000000;
    for(j=0;j<ciu;j++)
    {
        a=fun_esta_en_tour(I+1,j+1,vs);
        if(a==0 && matriz[verso-1][j]<d)
        {
            d=matriz[verso-1][j];
            verme=j+1;
        }
    }
    return(d);
}

int fun_nuevo_ver_so(FILE *ps,int ciu,int verso,int **matriz,
int I,int *vs,int *vcr,int dime)
{
    int a,j,veres,dimej,con,b;
    for(j=0;j<ciu;j++)
    {
        vcr[j]=0;
    }
    con=0;
    for(j=0;j<ciu;j++)
    {
        a=fun_esta_en_tour(I+1,j+1,vs);
        if(a==0 && matriz[verso-1][j]<=dime+dime)
        {
            vcr[con]=j+1;
            veres=vcr[con];
            dimej=matriz[verso-1][j];
            con++;
        }
    }
    b=fun_numero_aleatorio(0,con-1);
    return(vcr[b]);
}
```

```
int fun_esta_en_tour(int size,int verso,int *vs)
{
    int i;
    for(i=0;i<size;i++)
    {
    if(vs[i]==verso)
        return(1);
    }
    return(0);
}

void fun_escribe_solucion(FILE *ps,int ciu,int *vs,int **matriz)
{
    int i,suma;
    fprintf(ps,"\nSolución = ");
    for(i=0;i<ciu;i++)
    {
        fprintf(ps,"%d ",vs[i]);
    }
    suma=matriz[vs[ciu-1]-1][vs[0]-1];
    for(i=0;i<ciu-1;i++)
    {
        suma=suma+matriz[vs[i]-1][vs[i+1]-1];
    }
    fprintf(ps," Distancia = %d ",suma);
}

// 04.8-FUNCIONES DE LA FASE DE MEJORA DEL GRASP //
int fun_distancia(FILE *ps,int ciu,int *vs,int **matriz)
{
    int i,suma;
    suma=matriz[vs[ciu-1]-1][vs[0]-1];
    for(i=0;i<ciu-1;i++)
    {
        suma=suma+matriz[vs[i]-1][vs[i+1]-1];
    }
    return(suma);
```

```
}
void  funcion_copia_vector(FILE *ps,int ciu,int *vs,int *vss)
{
    int r;
    for(r=0;r<ciu;r++)
    {
        vss[r]=vs[r];
    }
}
void funcion_intercambio(FILE *ps,int inicio,int fin,int *vs)
{
    int aux;
    aux=vs[inicio];
    vs[inicio]=vs[fin];
    vs[fin]=aux;
}
void funcion_reorden(FILE *ps,int inicio,int fin,int *vs)
{
    int cota,aux,k;
    cota=(fin-inicio)/2;
    for(k=1;k<=cota;k++)
    {
        aux=vs[inicio-1+k];
        vs[inicio-1+k]=vs[fin-k];
        vs[fin-k]=aux;
    }
}
void escribe_vector(FILE *ps,int ciu,int *vs)
{
    int i;
    fprintf(ps,"\n\n");
    for(i=0;i<ciu;i++)
    {
        fprintf(ps,"%3d",vs[i]);
    }
}
// 04-FIN DE LAS FUNCIONES AUXILIARES //
```

15.3 Código C++ para el Algoritmo del Tabu Search

```
// 03-Algoritmo Tabu Search.cpp: define el punto de entrada de la
 aplicación de consola. //

// 01-INICIO DE LIBRERIAS //
#include "stdafx.h"
#include <malloc.h>
#include <stdio.h>
#include <tchar.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
# define getrandom(min,max)((rand()%(int)(((max)+1)-(min)))+(min))
// 01-FIN DE LIBRERÍAS //

// 02-INICIO DE CABECERA DE FUNCIONES //
void Interrumpir(char *texto);
void EscribeVector(FILE *ps,int dim,int *w);
int YaEsta(int size_vector_solucion,int nodo_a_buscar,int *v);
void SolucionVecinoMasCercano(int **m,int *v,int dim);
void EscribeMatriz(FILE *ps,int dim1,int dim2,int **matriz);
void ValorSolucion(FILE *ps,int **m,int dim, int *v, int *valor);
void LiberaMatriz(int **matriz,int dim);
int *ReservaVector(int dim);
void CopiaVector(int dim,int *v1, int *v2);
void Dos_permutacion(FILE *ps, int **m, int dim, int *xm, int *p,
 int *valor);
void Una_permutacion(FILE *ps, int **m, int dim, int *xm, int *p,
 int *valor);
void Todas_permutacion(FILE *ps, int **m, int dim, int *x, int *p,
 int *valor,int *mejor);
void IntroducirSolucionListaTabu(FILE *ps,int filas,int col,
int **LT,int *v,int **m,int dim);
void EscribeListaTabu(FILE *ps,int dim1,int dim2,int **LT);
int Esta_ListaTabu(int filas,int col,int **LT,int *p,int dim);
int Compara_Filas(int fila_i,int **LT,int *p,int dim);
```

```
int ValorPermutacion(int **m,int dim,int *p);
// 02-FIN DE CABECERA DE FUNCIONES //

// 03-INICIO DE FUNCIÓN PRINCIPAL, MAIN //
void main(int argc,char **argv)
{
    // 03.1-DECLARACIÓN DE VARIABLES //
    int dim,**m,*x,*p,*mejor,*mejor_tabu,**LT,**MT;
    int i,j,aux,valor,no_mejora,a,b,filas,col,sn,st,fmt,fn,h,fx;
    FILE *ps;
    FILE *pf;
    time_t tstart1,cstart1,tend1,cend1;

    // 03.2-ABRIR ARCHIVO DE ESCRITURA //
    ps=fopen("03-TABU.txt","w");
    if(ps==NULL)
        Interrumpir("PROBLEMAS AL RESERVAR MEMORIA");

    // 03.3-ABRIR ARCHIVO DE LECTURA //
    pf=fopen("gr17.txt","r");
    if(ps==NULL)
        Interrumpir("PROBLEMAS AL RESERVAR MEMORIA");
    fscanf(pf,"%d",&dim);
    fprintf(ps,"SOLUCIÓN DEL TSP-%d", dim);
    fprintf(ps," USANDO TABU SEARCH \n\n");
    fprintf(ps,"LA DIMENSIÓN ES %d\n\n", dim);

    // 03.4-RESERVA MATRIZ DINÁMICA //
    m=(int**)calloc(dim,sizeof(int*));
    if(m==NULL)
        printf("\n NO HAY ESPACIO EN MEMORIA PARA RESERVAR MATRIZ");

    // 03.5-LEE MATRIZ //
    for(i=0;i<dim;i++)
        m[i]=ReservaVector(dim);
    for(i=0;i<dim;i++)
    {
```

```
    fscanf(pf,"\n\n");
    for(j=0;j<dim;j++)
    fscanf(pf,"%d",&m[i][j]);
}
fclose(pf);

// 03.6-ESCRIBE MATRIZ //
EscribeMatriz(ps,dim,dim,m);
p=ReservaVector(dim);
mejor=ReservaVector(dim);
mejor_tabu=ReservaVector(dim);

// 03.7-RESERVA MATRIZ DINÁMICA //
filas=21;
col=dim+1;
LT =(int**)calloc(filas,sizeof(int*));
if(LT==NULL)
    printf("\n NO HAY ESPACIO EN MEMORIA PARA RESERVAR MATRIZ");
for(i=0;i<filas;i++)
    LT[i]=ReservaVector(col);
    MT =(int**)calloc(filas,sizeof(int*));
if(MT==NULL)
printf("\n NO HAY ESPACIO EN MEMORIA PARA RESERVAR MATRIZ");
for(i=0;i<filas;i++)
    MT[i]=ReservaVector(col);
    fprintf(ps,"\n\n\n");
//EscribeMatriz(ps,filas,col,LT);

// 03.8-INICIA RELOJ //
time(&tstart1);
cstart1=clock();

// 03.9-FASE CONSTRUCTIVA DEL TABU SEARCH //
fprintf(ps,"\n\n====================SOLUCION FASE
 CONSTRUCTIVA==========================\n\n");
x=ReservaVector(dim);
SolucionVecinoMasCercano(m,x,dim);
```

```
ValorSolucion(ps,m,dim,x,&fx);
fprintf(ps,"==================================================
==================\n");
st=ValorPermutacion(m,dim,x);
for(j=0;j<dim;j++)
MT[0][j]=x[j];
MT[0][dim]=st;
fprintf(ps,"\n\n\n <<<<<<<<<<<<<<<<<<<<  INICIO
  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>\n");
fprintf(ps,"\n SOLUCIÓN DE LA FASE CONSTRUCTIVA \n\n");
ValorSolucion(ps,m,dim,x,&valor);
EscribeListaTabu(ps,filas,col,MT);

// 03.10-FASE DE MEJORA DEL TABU SEARCH //
int CParada=1;
int esta;
fmt=100000;
while(CParada<=200)
{
    fprintf(ps,"\n ****************************************
    **********************\n");
    fprintf(ps,"\n --------------- ENTORNO DE SOLUCIONES
     NÚMERO  %d--------------------\n",CParada);
    CopiaVector(dim,x,p);
    Todas_permutacion(ps,m,dim,x,p,&valor,mejor);
    st=ValorPermutacion(m,dim,mejor);
    esta=0;
    for(i=0;i<filas;i++)
    {
        if(MT[i][dim]==st)
        {
            esta=1;
            i=filas+filas;
        }
    }
    if(esta==0)
    {
```

```
            for(j=0;j<dim;j++)
            {
                MT[CParada][j]=mejor[j];
                MT[CParada][dim]=st;
            }
        }
        EscribeListaTabu(ps,filas,col,MT);
        CopiaVector(dim,mejor,x);
        fn=ValorPermutacion(m,dim,mejor);
        if(fn<fmt)
        {
            fmt=fn;
            CopiaVector(dim,mejor,mejor_tabu);
        }
        CParada=CParada+1;
    }
    // FIN DEL WHILE //
    fprintf(ps,"\n\n ***** MEJOR SOLUCIÓN TABU SEARCH DESPUÉS
     DE %5d ITERACIONES \n",CParada-1);
    ValorSolucion(ps,m,dim,mejor_tabu,&valor);

    // 03.11-FINALIZA RELOJ //
    time(&tend1);
    cend1=clock();
    fprintf(ps,"\n TIEMPO = %4.3fseg/",
    ((float)cend1-cstart1)/CLK_TCK);

    // 03.12-LIBERA MEMORIA //
    free(x);
    free(p);
    free(mejor);
    free(mejor_tabu);
    LiberaMatriz(m,dim);
    LiberaMatriz(LT,filas);

    // 03.13-CIERRA FICHERA DE SALIDA //
    fclose(ps);
```

```
}
// 03-FINALIZA FUNCIÓN PRINCIPAL, MIN //

// 04-INICIAN FUNCIONES //

// 04.1-REALIZA TODAS LAS PERMUTACIONES //

void Todas_permutacion(FILE *ps, int **m, int dim, int *x,
 int *p, int *valor,int *mejor)
{
    int i,aux,j,a,b,c,d,e,*o,*q;
    fprintf(ps,"\n\n");
    o=ReservaVector(dim);
    q=ReservaVector(dim);
    //fprintf(ps,"\n Entorno N(x)-> Todas_Permutaciones \n");
    CopiaVector(dim,x,p);
    ValorSolucion(ps,m,dim,p,valor);
    //fprintf(ps,"\n Todas_Permutación \n");
    b=100000;
    d=100000;
    for(i=0;i<=dim-3;i++)
    {
        for(j=i+1;j<=dim-2;j++)
        {
            aux=p[i+1];
            p[i+1]=p[j+1];
            p[j+1]=aux;
            ValorSolucion(ps,m,dim,p,valor);
            a=ValorPermutacion(m,dim,p);
            if(a<b)
            {
                b=a;
                CopiaVector(dim,p,o);
            }
            CopiaVector(dim,x,p);
        }
    }
```

```
    fprintf(ps,"\n PRIMER FOR \n");
    ValorSolucion(ps,m,dim,o,&e);
    fprintf(ps,"\n FIN DEL PRIMER FOR \n");
    for(i=0;i<dim;i++)
        mejor[i]=0;
    for(j=0;j<=dim-2;j++)
    {
        aux=p[0];
        p[0]=p[j+1];
        p[j+1]=aux;
        ValorSolucion(ps,m,dim,p,valor);
        c=ValorPermutacion(m,dim,p);
        if(c<d)
        {
            d=c;
            CopiaVector(dim,p,q);
        }
        CopiaVector(dim,x,p);
    }
    fprintf(ps,"\n SEGUNDO FOR \n");
    ValorSolucion(ps,m,dim,q,&e);
    fprintf(ps,"\n FIN DEL SEGUNDO FOR \n");
    if(b<d)
        CopiaVector(dim,o,mejor);
    else
        CopiaVector(dim,q,mejor);
        fprintf(ps,"\n MEJOR SOLUCIÓN DEL ENTORNO \n");
        ValorSolucion(ps,m,dim,mejor,&e);
    fprintf(ps,"\n ----------------------------------------------
    --------------------\n");
    fprintf(ps,"\n\n");
}

// 04.2-INTRODUCE SOLUCIÓN EN LISTA TABÚ //
void IntroducirSolucionListaTabu(FILE *ps,int filas,int col,
int **LT,int *v,int **m,int dim)
{
```

```
    //EscribeVector(ps,dim,v);
    int i,j,cont,a;
    cont=0;
    for(i=0;i<filas;i++)
    {
        if(LT[i][col-1]!=0)
        cont++;
    }
    a=ValorPermutacion(m,dim,v);
    LT[cont][col-1]=a;
    for(j=0;j<col-1;j++)
    {
        LT[cont][j]=v[j];
    }
}

// 04.3-CALCULA VALOR DE LA PERMUTACIÓN //
int ValorPermutacion(int **m,int dim,int *p)
{
    int i,valor;
    valor=m[p[0]-1][p[dim-1]-1];
    for(i=0;i<dim-1;i++)
    {
        valor=valor + m[p[i]-1][p[i+1]-1];
    }
    return (valor);
}

// 04.4-COMPARA FILAS //
int Compara_Filas(int fila_i,int **LT,int *p,int dim)
{
    int j,contenido;
    for(j=0;j<dim;j++)
    {
        if(LT[fila_i][j]=p[j])
            return(0);
        else
```

```
            contenido=1;
    }
    if(contenido==1)
        return(1);
}

// 04.5-REVISA SI ESTÁ EN LISTA TABÚ //
int Esta_ListaTabu(int filas,int col,int **LT,int *p,int dim)
{
    int i,no_esta,La_Misma;
    for(i=0;i<filas;i++)
    {
        if(LT[i][0]!=0)
        {
            La_Misma=Compara_Filas(i,LT,p,dim);
            if(La_Misma==0)
            {
                no_esta=0;
            }
            else
                return(1);
        }
        else
            i=filas+filas;
    }
    if(no_esta==0)
        return(0);
}

// 04.6-CALCULA EL VALOR DE LA SOLUCIÓN //
void ValorSolucion(FILE *ps,int **m,int dim, int *v, int *valor)
{
    int i;
    *valor=m[v[0]-1][v[dim-1]-1];
    for(i=0;i<dim-1;i++)
    {
        *valor=*valor+m[v[i]-1][v[i+1]-1];
```

```
    }
    for(i=0;i<dim;i++)
    {
        fprintf(ps,"%6d",v[i]);
    }
    fprintf(ps,"   VALOR %6d \n",*valor);
}

// 04.7-ESCRIBE LISTA TABÚ //
void EscribeListaTabu(FILE *ps,int dim1,int dim2,int **LT)
{
    int i,j;
    fprintf(ps,"\n\n");
    fprintf(ps,"LISTA TABÚ \n");
    for(i=0;i<dim1;i++)
    {
        for(j=0;j<dim2-1;j++)
        {
            fprintf(ps,"%5d",LT[i][j]);
        }
        fprintf(ps,"       VALOR %6d", LT[i][dim2-1]);
        fprintf(ps,"\n\n");
    }
    }

// 04.8-REALIZA DOS PERMUTACIONES //
void Dos_permutacion(FILE *ps, int **m, int dim, int *xm, int *p,
 int *valor)
{
    int i,j,aux1, aux2;
    fprintf(ps,"\n 2_PERMUTACIÓN \n");
    CopiaVector(dim,xm,p);
    for(i=0;i<dim-3;i++)
    {
        aux1=p[i];
        p[i]=p[i+1];
        p[i+1]=aux1;
```

```
        aux2=p[i+2];
        p[i+2]=p[i+2+1];
        p[i+2+1]=aux2;
        ValorSolucion(ps,m,dim,p,valor);
        CopiaVector(dim,xm,p);
        fprintf(ps,"\n\n");
    }
}

// 04.9-REALIZA UNA PERMUTACIÓN //
void Una_permutacion(FILE *ps, int **m, int dim, int *xm, int *p,
 int *valor)
{
    int i,aux;
    ValorSolucion(ps,m,dim,p,valor);
    fprintf(ps,"\n 1_PERMUTACIÓN \n");
    CopiaVector(dim,xm,p);
    for(i=0;i<dim-1;i++)
    {
        aux=p[i];
        p[i]=p[i+1];
        p[i+1]=aux;
        ValorSolucion(ps,m,dim,p,valor);
        CopiaVector(dim,xm,p);
        fprintf(ps,"\n\n");
    }
}

// 04.10-COPIA VECTOR O COPIA PERMUTACIÓN //
void CopiaVector(int dim,int *v1, int *v2)
{
    int i;
    for(i=0;i<dim;i++)
    {
        v2[i]=v1[i];
    }
}
```

```
// 04.11-RESERVA VECTOR //
int *ReservaVector(int dim)
{
    int *v;
    v=(int*)calloc(dim,sizeof(int));
    if(v==NULL)
        printf("PROBLEMAS EN LECTURA");
    return(v);
}

// 04.12-LIBERA MATRIZ //
void LiberaMatriz(int **matriz,int dim)
{
    int i;
    for(i=0;i<dim;i++)
    {
        free(matriz[i]);
    }
    free(matriz);
}

// 04.13-ESCRIBE MATRIZ //
void EscribeMatriz(FILE *ps,int dim1,int dim2,int **matriz)
{
    int i,j;
    fprintf(ps,"MATRIZ DE %d FILAS Y %d COLUMNAS \n\n",dim1,dim2);
    for(i=0;i<dim1;i++)
    {
        for(j=0;j<dim2;j++)
        {
            fprintf(ps,"%6d",matriz[i][j]);
        }
        fprintf(ps,"\n\n");
    }
}
```

```
// 04.14-INTERRUMPIR //
void Interrumpir(char *texto)
{
    printf("\n%s",texto);
    exit(0);
}

// 04.15-ESCRIBE VECTOR O PERMUTACIÓN //
void EscribeVector(FILE *ps,int dim,int *w)
{
    int i;
    fprintf(ps,"\n\n");
    for(i=0;i<dim;i++)
    {
        fprintf(ps,"%4d",w[i]);
    }
    fprintf(ps,"\n\n");
}

// 04.16-ENCUENTRA EL NUEVO NODO MÁS CERCANO //
void SolucionVecinoMasCercano(int **m,int *v,int dim)
{
    int j,ye,r,k,ind;
    long int min;
    j=getrandom(1,dim);
    v[0]=j;
    for(r=1;r<dim;r++)
    {
        min=100000;
        for(k=0;k<dim;k++)
        {
            ye=YaEsta(r,k+1,v);
            if(ye==0 && m[j-1][k]!=0 && m[j-1][k]<min)
            {
                min=m[j-1][k];
                ind=k;
            }
```

```
        }
        j=ind+1;
        v[r]=j;
    }
}

// 04.17-BUSCA SI EL NUEVO NODO YA ESTÁ EN EL VECTOR O PERMUTACIÓN //
int YaEsta(int size_vector_solucion,int nodo_a_buscar,int *v)
{
    int i;
    for(i=0;i<size_vector_solucion;i++)
    {
        if(v[i]==nodo_a_buscar)
            return(1);
    }
    return(0);
}
// 04-FINALIZAN FUNCIONES AUXILIARES //
```

15.4 Código C++ para el Algoritmo del Scatter Search

15.4.1 Main del Scatter Search

```
// 04-Algoritmo Scatter Search.cpp: define el punto de entrada
 de la aplicación de consola. //

#include "tipos.h"
int tam_pob=0;
int tam_ref=0;
int evafobjetivo=0;
void main(int argc,char **argv)
{
    int ciudades,**matriz,*vso,*vsop,**poblacion,**conj_ref,k,
    *aux,**mezcla,tam_mezcla=0;
    int register i;
    int sigue=1,contador=0,niter=0,contsigue=0,no_nueva=0;
    // pudo actualizar el conj_ref =1 else 0
    time_t tstart1,cstart1,tend1,cend1,tstartM,cstartM,tendT,cendT;
    tam_pob=atoi(argv[2]);
    tam_ref=atoi(argv[3]);
    FILE *ps;
    ps=fopen("scatter.txt","a");
    if(ps==NULL)
    {
        printf("no hay espacio en memoria para escribir resultados");
        exit(0);
    }

    // fprintf(ps,"\n%s\n\n",argv[1]);
    matriz=fun_leer_matriz_dis(&ciudades,argv[1]);
    // fun_escribe_matriz_dis(ps,ciudades,ciudades,matriz);

    vso=fun_reserva_vector(ciudades);
    vsop=fun_reserva_vector(ciudades);
    aux=fun_reserva_vector(ciudades);
    poblacion=fun_reserva_matriz_dis(tam_pob,ciudades+1);
    conj_ref=fun_reserva_matriz_dis(tam_ref,ciudades+2);
```

```
for(i=0;i<tam_ref;i++)
    tam_mezcla+=i;
    mezcla=fun_reserva_matriz_dis(tam_mezcla,ciudades+1);

time(&tstart1);
cstart1=clock();

for(k=0;k<tam_pob && no_nueva<No_nueva &&
 ((float)cend1-cstart1)/CLK_TCK<0.4*ciudades ;k++)
{
    // FASE CONSTRUCTIVA
    // fprintf(ps,"\nSOLUCION INICIAL");
    fun_solucion_inicial(ciudades,vso,matriz);
    // fun_escribe_solucion(ps,ciudades,vso,matriz);

    // FASE DE MEJORA
    // fprintf(ps,"\nSOLUCION MEJORADA");
    fun_solucion_mejorada(ciudades,vso,vsop,matriz);
    // fun_escribe_solucion(ps,ciudades,vsop,matriz);
    // fprintf(ps,"\n\n");

    // mirar si ya esta esta solucion
    if (fun_es_igual(ciudades,vsop,poblacion,k-1)==0)
    {
        // CONSTRUYE POBLACION
        fun_poblacion(ciudades,vsop,matriz,poblacion);
        no_nueva=0;
    }
    else
    {
        no_nueva++;
        k--;
    }
    time(&tend1);
    cend1=clock();
}
if (no_nueva==No_nueva)
```

```
{
    if (k<tam_pob) (tam_ref=(k/8)*2);
    (tam_pob=k);
}
if (((float)cend1-cstart1)/CLK_TCK>0.4*ciudades) tam_pob=k;
    // Pinta matriz poblacion
    // fun_escribe_matriz_pob(ps,tam_pob,ciudades,poblacion);
    // ORDENA POBLACION en orden ascendente
    fun_ordena_poblacion(ps,ciudades,poblacion,tam_pob);
    // CREAR CONJUNTO DE REFERENCIA
    // Metemos en referencia los tam_ref/2 mejores de la
     poblacion
    fun_meter_enreferencia(ciudades,poblacion,conj_ref);
    // metemos los restantes diversos
    fun_meter_diversos(ciudades,poblacion,conj_ref);
    // fun_escribe_matriz_pob(ps,tam_pob,ciudades,poblacion);
    // fun_escribe_matriz_pob(ps,tam_ref,ciudades,conj_ref);
    // Mientras no pase contador y se vaya actualizando el
     cojunto de referencia
    time(&tstartM);
    cstartM=clock();
    while(sigue!=0 && contador<1000 &&
     ((float)cend1-cstartM)/CLK_TCK<1.75*ciudades)
    {
        // Hacer mezcla de poblacion
        // fun_escribe_matriz_mez(ps,tam_ref,ciudades+1,
         conj_ref);
        tam_mezcla=fun_mezcla_poblacion(ciudades,conj_ref,
         mezcla,matriz,niter);
        // fun_escribe_matriz_mez(ps,tam_mezcla,ciudades+1,
         mezcla);
        // Hacer mejora local de toda la poblacion
        // fprintf(ps,"Tam_mezcla%d\n",tam_mezcla);
        fun_mejora_mezcla(ciudades,mezcla,matriz,tam_mezcla);
        // Ordenamos mezcla
        fun_ordena_poblacion(ps,ciudades,mezcla,tam_mezcla);
        // Ordenamos conjunto de referencia
```

```
            fun_ordena_poblacion(ps,ciudades,conj_ref,tam_ref);
            // Actualización del conjunto de referencia
            // Si sigue=0 no se ha metido nada mejor
            niter++;
            // fun_escribe_matriz_mez(ps,tam_mezcla,ciudades+2,
             mezcla);
            sigue=fun_meter_mezcla(ciudades,mezcla,conj_ref,
             tam_mezcla,niter);
            contador+=1;
            // fun_escribe_matriz_mez(ps,tam_ref,ciudades+2,
             conj_ref);
            time(&tend1);
            cend1=clock();
        }
        time(&tendT);
        cendT=clock();
        fprintf(ps,"Instancia\t%s\tTiempoG\t%4.2f\tTiempoM\
         t%4.2f\tTiempoT\t%4.2f\tPob_ini\t%d\tValor\t%d\tEvaluaciones\t%d\t
          CLK_TCK,((float)cendT-cstartM)/CLK_TCK,
           ((float)cendT-cstart1)/CLK_TCK,tam_pob,
            conj_ref[0][ciudades],evafobjetivo,tam_ref);
        // Escribir resultados
        // fun_escribe_matriz_mez(ps,tam_mezcla,ciudades+1,mezcla);
        // fun_escribe_matriz_mez(ps,tam_ref,ciudades+1,conj_ref);
        // conj_ref[2]=fun_mezclar_soluciones(ciudades,
         conj_ref[0], conj_ref[1]);
        // fun_escribe_matriz_pob(ps,tam_ref,ciudades,conj_ref);
        fclose(ps);
}
```

15.4.2 Fase Constructiva

```
#include "tipos.h"
extern int tam_pob;
extern int tam_ref;
//FASE CONSTRUCTIVA
void fun_solucion_inicial(int ciudades,int *vso,int **matriz)
```

```
{
    int i,ver_az,ver_so,di_me;
    int *vca,*vcar;
    vca=fun_reserva_vector(ciudades-1);
    vcar=fun_reserva_vector(ciudades);
    ver_az=fun_numero_aleatorio(1,ciudades);
    ver_so=ver_az;
    fun_ver_candidatos(ciudades,ver_so,vca);
    for(i=0;i<ciudades-1;i++)
    {
        vso[i]=ver_so;
        di_me=fun_verdime(ciudades,ver_so,matriz,i,vso);
        ver_so=fun_nuevo_ver_so(ciudades,ver_so,matriz,i,vso,
         vcar,di_me);
        vso[i+1]=ver_so;
    }

}

// funcion vertices candidatos
void fun_ver_candidatos(int ciu,int verso ,int *vc)
{
    int i,j;
    j=0;
    for(i=0;i<ciu;i++)
    {
        vc[i]=i+1;
        if(vc[i]!=verso)
        {
            vc[j]=i+1;
            j++;
        }
    }
}

//funcion vertice y distancia mejor
int fun_verdime(int ciu,int verso,int **matriz,int I,int *vs)
```

```
{
    int a,j,verme,d=100000;
    for(j=0;j<ciu;j++)
    {
        a=fun_esta_en_tour(I+1,j+1,vs);
        if(a==0 && matriz[verso-1][j]<d)
        {
            d=matriz[verso-1][j];
            verme=j+1;
        }
    }
    return(d);
}

//funcion nuevo vertice solucion
int fun_nuevo_ver_so(int ciu,int verso,int **matriz,int I,
 int *vs,int *vcr,int dime)
{
    int a,j,veres,dimej,con,b;
    for(j=0;j<ciu;j++)
    {
        vcr[j]=0;
    }
    con=0;
    for(j=0;j<ciu;j++)
    {
        a=fun_esta_en_tour(I+1,j+1,vs);
        if(a==0 && matriz[verso-1][j]<=dime+dime)
        {
            vcr[con]=j+1;
            veres=vcr[con];
            dimej=matriz[verso-1][j];
            con++;
        }
    }
    b=fun_numero_aleatorio(0,con-1);
    return(vcr[b]);
```

```
}

//funcion esta en tour?
int fun_esta_en_tour(int size,int verso,int *vs)
{
    int i;
    for(i=0;i<size;i++)
    {
        if(vs[i]==verso)
        return(1);
    }
    return(0);
}

//funcion escribe solucion
void fun_escribe_solucion(FILE *ps,int ciu,int *vs,int **matriz)
{
    int i,dist;
    fprintf(ps,"\nsolucion = ");
    for(i=0;i<ciu;i++)
    fprintf(ps,"%d ",vs[i]);
    dist=fun_distancia(ciu,vs,matriz);
    fprintf(ps,"distancia = %d ",dist);
}
```

15.4.3 Fase de Mejora

```
#include "tipos.h"
extern int tam_pob;
extern int tam_ref;
void fun_solucion_mejorada(int ciudades,int *vso,int *vsop,
 int **matriz)
{
    int i,j,dfm,daux,contador,costea,a,b,c,d,antiguo,nuevo,coste;
    int *vsoa,*vsoaa,mejorada=0;
    vsoa=fun_reserva_vector(ciudades);
    vsoaa=fun_reserva_vector(ciudades);
```

```
// for(i=0;i<ciudades;i++)
    // printf("%d  ",vso[i]);
    // printf("\n");
funcion_copia_vector(ciudades,vso,vsop);
dfm=fun_distancia(ciudades,vsop,matriz);
daux=10000000;
contador=1;
while(dfm<daux)
{
    costea=10000000;
    for(i=0;i<ciudades-2;i++)
    {
        a=vsop[i];
        b=vsop[i+1];
        for(j=i+2;j<ciudades;j++)
        {
            c=vsop[j];
            if(j+1<ciudades) d=vsop[j+1];
            else d=vsop[0];
            antiguo=matriz[a-1][b-1]+matriz[c-1][d-1];
            nuevo=matriz[a-1][c-1]+matriz[b-1][d-1];
            if(nuevo<antiguo)
            {
                coste=fun_distancia(ciudades,vsop,matriz)-
                antiguo+nuevo;
                if(coste<costea)
                {
                    costea=coste;
                    mejorada=1;
                    funcion_copia_vector(ciudades,vsop,vsoa);
                    funcion_intercambio(i+1,j,vsoa);
                    if(j-(i+1)<=2)
                    funcion_copia_vector(ciudades,vsoa,vsoaa);
                    else
                    {
                        funcion_reorden(i+2,j,vsoa);
                        funcion_copia_vector(ciudades,vsoa,
```

```
                                vsoaa);
                            }
                        }
                    }
                    //else
                    //  funcion_copia_vector(ciudades,vso,vsoaa);
                }
            }
            if (mejorada==1)
            {
                daux=dfm;
                // for(i=0;i<ciudades;i++)
                // printf("%d  ",vsoaa[i]);
                // printf("\n");
                dfm=fun_distancia(ciudades,vsoaa,matriz);
                if(dfm<daux)
                {
                    // fprintf(ps,"\nmejora %d",contador);
                    // fun_escribe_solucion(ps,ciudades,vsoaa,matriz);
                    funcion_copia_vector(ciudades,vsoaa,vsop);
                }
                contador++;
            }
            else
            {
                funcion_copia_vector(ciudades,vso,vsop);
                dfm=daux;
            }
        }
}

//funcion que hace intercambio
void funcion_intercambio(int inicio,int fin,int *vs)
{
    int aux;
    aux=vs[inicio];
    vs[inicio]=vs[fin];
```

```
    vs[fin]=aux;
}

//funcion que reordena las componenetes del vector
void funcion_reorden(int inicio,int fin,int *vs)
{
    int cota,aux,k;
    cota=(fin-inicio)/2;
    for(k=1;k<=cota;k++)
    {
        aux=vs[inicio-1+k];
        vs[inicio-1+k]=vs[fin-k];
        vs[fin-k]=aux;
    }
}
```

15.4.4 Función Distancia

```
#include "tipos.h"
/*
dim: numero de ciudades
vs : vector con permutacion
m : matriz
*/

//funcion que calcula la distancia segun la permutacion
int fun_distancia(int ciu,int *vs,int **m)
{
    int i,suma;
    // evafobjetivo++;
    suma=m[vs[ciu-1]-1][vs[0]-1];
    for(i=0;i<ciu-1;i++)
    suma=suma+m[vs[i]-1][vs[i+1]-1];
    return(suma);
}
```

15.4.5 Función Lee Datos

```
#include "tipos.h"
/*
dim: numero de ciudades
vs : vector con permutacion
m : matriz
*/

//funcion que calcula la distancia segun la permutacion
int fun_distancia(int ciu,int *vs,int **m)
{
    int i,suma;
    // evafobjetivo++;
    suma=m[vs[ciu-1]-1][vs[0]-1];
    for(i=0;i<ciu-1;i++)
    suma=suma+m[vs[i]-1][vs[i+1]-1];
    return(suma);
}
```

15.4.6 Función Población

```
#include "tipos.h"
extern int tam_pob;
extern int tam_ref;
int r=0;

//funcion que construye la poblacion
void fun_poblacion(int ciu,int *vs,int **m,int **p)
{
    int j,dist;
    for(j=0;j<ciu;j++)
    p[r][j]=vs[j];
    dist=fun_distancia(ciu,vs,m);
    p[r][ciu]=dist;
    r++;
}
```

```
//funcion escribe matriz de poblacion
void fun_escribe_matriz_pob(FILE *ps,int dim1,int dim2,int **p)
{
    int i,j;
    fprintf(ps,"\n\npoblacion\n");
    for(i=0;i<dim1;i++)
    {
        for(j=0;j<=dim2;j++)
        {
            if(j<dim2)
            fprintf(ps,"%4d",p[i][j]);
            else
                fprintf(ps,"%8d",p[i][j]);
        }
        fprintf(ps,"\n");
    }
}
```

15.4.7 Función Útiles

```
#include "tipos.h"
#include <stdlib.h>
extern int tam_pob;
extern int tam_ref;
extern evafobjetivo;
//funcion numero aleatorio
int fun_numero_aleatorio(int m,int n)
{
    int numero_aleatorio;
    numero_aleatorio=getrandom(m,n);
    return(numero_aleatorio);
}
//funcion copia vector
void  funcion_copia_vector(int ciu,int *vs,int *vss)
{
    int r;
    for(r=0;r<ciu;r++)
```

```
    {
        vss[r]=vs[r];
    }
}
//funcion que ordena las permutaciones segun el valor de la
 funcion objetivo
void fun_ordena_poblacion(FILE *ps,int ciudades,int **p,int tam_mat)
{
    int i,j,aux,*va,*va1,*va2;
    va=fun_reserva_vector(ciudades);
    va1=fun_reserva_vector(ciudades);
    va2=fun_reserva_vector(ciudades);
    for(i=0;i<tam_mat-1;i++)
    {
        for(j=0;j<tam_mat-1;j++)
        {
            if((p[j][ciudades])>(p[j+1][ciudades]))
            {
                aux=p[j][ciudades];
                p[j][ciudades]=p[j+1][ciudades];
                p[j+1][ciudades]=aux;
                fun_copia_matriz_vector(j,ciudades,va,p);
                fun_copia_matriz_vector(j+1,ciudades,va1,p);
                fun_copia_vector_matriz(j,ciudades,va1,p);
                fun_copia_vector_matriz(j+1,ciudades,va,p);
            }
        }
    }
// fun_escribe_vector(ps,tam_pob,v);
}

void fun_copia_matriz_vector(int fila,int ciudades,int *v,int **p)
{
    int k;
    for(k=0;k<ciudades;k++)
    v[k]=p[fila][k];
}
```

```
void fun_copia_vector_matriz(int fila,int ciudades,int *v,int **p)
{
    int k;
    for(k=0;k<ciudades;k++)
    p[fila][k]=v[k];
}
//funcion escribe vector
void fun_escribe_vector(FILE *ps,int dim_vector,int *vector)
{
    int i;
    fprintf(ps,"\n");
    for(i=0;i<dim_vector;i++)
    fprintf(ps,"%8d",vector[i]);
    fprintf(ps,"\n");
}
// funcion que mete en conjunto de referencia los tam_ref/2
 mejores de la poblacion que se supone ya ordenada
void fun_meter_enreferencia(int ciudades, int **MATRIZ1,
 int **MATRIZ2)
{
    int a=(tam_ref/2);
    register int i,j;
    for (i=0;i<a;i++)
    {
        for (j=0;j<=ciudades;j++)
        MATRIZ2[i][j]=MATRIZ1[i][j];
    }
}
//funcion que haya los más diversos de la población y los mete
 en el conjunto de referencia
void fun_meter_diversos(int ciudades, int **MATRIZ1, int ** MATRIZ2)
{
    int Cota_min=0,Cota_max,n_diversos=0;//n_divesos:
     numero de diversos que hemos metido
    int elegido=0,*ant_elegidos,temp1=0,temp2=0,temp=0;
    int register i,j;
```

```
    ant_elegidos=fun_reserva_vector(tam_ref/2);
    for (n_diversos=0;n_diversos<tam_ref/2;n_diversos++)
    {
        Cota_min=0;
        for (i=tam_ref/2;i<tam_pob ;i++)
        {
            if (no_esta_elegido(ant_elegidos,i,tam_ref/2)==0)
            {
                Cota_max=99999999;
                for(j=0;j<(tam_ref/2)+n_diversos;j++)
                {
                    temp=distancia(ciudades,MATRIZ1[i],MATRIZ2[j]);
                    if (temp<Cota_max) Cota_max=temp;
                }
                if (Cota_max>Cota_min)
                {
                    Cota_min=Cota_max;
                    elegido=i;
                }
                // printf("%d\n",Cota_min);
            }
        }
        // printf("%d\n",Cota_min);
        ant_elegidos[n_diversos]=elegido;
        meter_elegido(ciudades,MATRIZ1[elegido],(tam_ref/2)+
        n_diversos,MATRIZ2);
    }
}
// funcion que nos dice si ya hemos metido un elemento
int no_esta_elegido(int *vector1,int num,int contador)
{
    int register i;
    for(i=0;i<contador;i++)
    {
        if (vector1[i]==num)
        return 1;
    }
```

```
    return 0;
}
//funcion que mete el más diverso dentro del conjunto de referencia
void meter_elegido(int ciudades,int *vector1,int posicion,
int **MATRIZ2)
{
    int register i;
    for(i=0;i<=ciudades;i++)
    MATRIZ2[posicion][i]=vector1[i];
}
//funcion que calcula la distancia de una permutacion
int distancia(int ciudades,int *vector1, int *vector2)
{
    int register i,j;
    int temp1=0,temp2=0,temp=0;
    for (i=0;i<ciudades;i++)
    {
        temp1=i;
        for (j=0;j<ciudades;j++)
        {
            temp2=j;
            if (vector2[j]==vector1[i]) break;
        }
        temp+=abs(temp1-temp2);
    }
    return temp;
}
//funcion que mezcla dos soluciones y nos devuelve la solucion
 mezclada
int  *fun_mezclar_soluciones(int ciudades,int *vector1,
int *vector2)
{
    int *vector_nuevo;
    int temp1=0,temp2=0,caso=0;
    int register i;
    int p=0;
    vector_nuevo=fun_reserva_vector(ciudades);
```

```
for(i=0;i<ciudades;i++)
{
    if (i==0)
    {
        if (vector1[i]!=vector2[i])
        {
            p=getrandom(0,10);
            if (p>=5)
            {
                vector_nuevo[i]=vector1[i];
                temp1=fun_indice_siguiente(ciudades,
                vector_nuevo,vector1);
            }
            else
            {
                vector_nuevo[i]=vector2[i];
                temp2=fun_indice_siguiente(ciudades,
                vector_nuevo,vector2);
            }
        }
        else
        {
            vector_nuevo[i]=vector1[i];
            temp1=fun_indice_siguiente(ciudades,
            vector_nuevo,vector1);
            temp2=fun_indice_siguiente(ciudades,
            vector_nuevo,vector2);
        }
    }
    else
    {
        if (vector1[temp1]!=vector2[temp2])
        {
            if (temp1==temp2) caso=0;
            else caso=(temp1-temp2)/(abs(temp1-temp2));
                switch(caso)
            {
```

```
                    case -1:
                    vector_nuevo[i]=vector1[temp1];
                    temp1=fun_indice_siguiente(ciudades,
                    vector_nuevo,vector1);
                    break;
                    case 1:
                    vector_nuevo[i]=vector2[temp2];
                    temp2=fun_indice_siguiente(ciudades,
                    vector_nuevo,vector2);
                    break;
                    case 0:
                    p=getrandom(0,10);
                    if (p>=5)
                    {
                        vector_nuevo[i]=vector1[temp1];
                        temp1=fun_indice_siguiente(ciudades,
                        vector_nuevo,vector1);
                    }
                    else
                    {
                        vector_nuevo[i]=vector2[temp2];
                        temp2=fun_indice_siguiente(ciudades,
                        vector_nuevo,vector2);
                    }
                    break;
                }
            }
            else
            {
                vector_nuevo[i]=vector1[temp1];
                temp1=fun_indice_siguiente(ciudades,
                vector_nuevo,vector1);
                temp2=fun_indice_siguiente(ciudades,
                vector_nuevo,vector2);
            }
        }
    }
```

```
    return vector_nuevo;
}
//funcion necesaria para hacer la mezcla nos da el primer indice
 que no esta colocado en la permutacion mezcla
int fun_indice_siguiente(int ciudades,int * vector_mezcla,
 int * vector_orig)
{
    //funcion que me dice cual es el primer indice del segundo
    vector que no que no esta en el primero
    int register i;
    for (i=0;i<ciudades;i++)
    {
        if (no_esta_elegido(vector_mezcla,vector_orig[i],ciudades)==0)
        {
            return i;
        }
    }
    return 0;
}
//funcion que hace toda la mezcla del conjunto de referencia
int fun_mezcla_poblacion(int ciudades, int **conj_ref,
int **mezcla,int **matriz, int niter)
{
    int posicion=0,*aux;
    int register j,k;
    aux=fun_reserva_vector(ciudades);
    for (j=0;j<tam_ref;j++)
    {
        for(k=j+1;k<tam_ref;k++)
        {
            if (conj_ref[j][ciudades+1]==niter ||
             conj_ref[k][ciudades+1]==niter)
            {
                aux=fun_mezclar_soluciones(ciudades,conj_ref[j],
                conj_ref[k]);
                funcion_copia_vector(ciudades,aux,mezcla[posicion]);
                mezcla[posicion][ciudades]=fun_distancia(ciudades,
```

```
                mezcla[posicion],matriz);
                posicion+=1;
            }
        }
    }
    return posicion;
}
//funcion que  hace la mejora local de toda la mezcla
void fun_mejora_mezcla(int ciudades, int **mezcla,int **matriz,
int tam_mezcla)
{
    int posicion=0;
    int register j,k;
    for (posicion=0;posicion<tam_mezcla;posicion++)
    {
        fun_solucion_mejorada(ciudades,mezcla[posicion],
        mezcla[posicion],matriz);
        mezcla[posicion][ciudades]=fun_distancia(ciudades,
        mezcla[posicion],matriz);
        posicion+=1;
    }
}
//funcion que actuliza el conjunto de referencia metiendo los
 mejores de mezcla, la mezcla y el conjunto de referencia
  deben estar ordenados al actualizar si alguno ya estaba
   metido no lo mete
int fun_meter_mezcla(int ciudades,int **mezcla,int **conj_ref,
int tam_mezcla,int niter)
{
    //esta funcion devuelve 1 si ha mejorado alguna
    //y 0 en otro caso;
    int register i,j;
    int ninguno=1,posicion=0;
    for (i=0;i<tam_mezcla;i++)
    {
        posicion=-1;
        for (j=tam_ref-1;j>=0;j--)
```

```
{
    if ((mezcla[i][ciudades]<conj_ref[j][ciudades]))
    {
        posicion=j;
        ninguno=0;
    }
}
if (posicion!=-1)
{
    if (posicion!=0)
    {
        if (mezcla[i][ciudades]!=conj_ref[posicion-1]
        [ciudades])
        {
            funcion_copia_vector(ciudades+1,mezcla[i],
            conj_ref[posicion]);
            conj_ref[posicion][ciudades+1]=niter;
        }
        else
        {
            if (fun_es_igual(ciudades,mezcla[i],conj_ref,
            posicion-1)!=1)
            {
                funcion_copia_vector(ciudades+1,
                mezcla[i],conj_ref[posicion]);
                conj_ref[posicion][ciudades+1]=niter;
            }
            else
                ninguno=1;
        }
    }
    else
        funcion_copia_vector(ciudades+1,mezcla[i],
        conj_ref[posicion]);
        conj_ref[posicion][ciudades+1]=niter;
}
else
```

```
        {
            if (ninguno==1)
            {
                return 0;
            }
            else
            {
                return 1;
            }
        }
    }
    return 0;
}
// mira si una permutacion esta en las posi primeras fila de
 mezcla, devuelve 0 si no es igual y 1 si es igual
int fun_es_igual(int ciudades, int *vector1,int **mezcla,int posi)
{
    int register i=0,j;
    int no_igual=0;
    for (i=posi;i>=0;i--)
    {
        no_igual=0;
        for(j=0;j<ciudades;j++)
        {
            if (vector1[j]!=mezcla[i][j])
            no_igual=1;
        }
        if (no_igual==0) return 1;
    }
    return 0;
}
```

SOLUCIÓN DEL BRAZIL58 USANDO INDICE Y SOLVER

58

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	2635	2713	2437	1600	2845	6002	1743	594	2182	2906	1658	464
2	2635	0	314	2636	666	1096	4645	693	2889	287	772	1135	2875
3	2713	314	0	2730	706	791	4588	922	2991	217	760	1050	2915
4	2437	2636	2730	0	2824	3457	6083	2120	2040	2844	2973	3937	1958
5	1600	666	706	2824	0	1247	4746	716	2223	584	1442	694	2089
6	2845	1096	791	3457	1247	0	5126	1653	3545	708	1298	1342	3451
7	6002	4645	4588	6083	4746	5126	0	4267	6553	4826	3830	5951	6417
8	1743	693	922	2120	716	1653	4267	0	2296	923	1175	1282	2162
9	594	2889	2991	2040	2223	3545	6553	2296	0	2869	3459	2366	147
10	2182	287	217	2844	584	708	4826	923	2869	0	1004	781	2793
11	2906	772	760	2973	1442	1298	3830	1175	3459	1004	0	2128	3322
12	1658	1135	1050	3937	694	1342	5951	1282	2366	781	2128	0	2265
13	464	2875	2915	1958	2089	3451	6417	2162	147	2793	3322	2265	0
14	3334	1424	1119	3710	1577	347	5423	1906	3741	1036	1609	1695	3607
15	3987	2185	1776	4363	2230	1126	6089	2559	4399	1694	2266	2353	4265
16	2870	1193	1451	2941	1609	1989	3142	1133	3417	1694	693	2819	3283
17	2601	846	5410	3207	1003	250	4876	1352	3163	458	1041	1117	3029
18	330	2142	2182	1991	1260	2736	5676	1422	924	1844	2585	1557	931
19	3049	1127	846	3512	1297	138	5181	1651	3462	757	1353	1416	3328
20	1302	3104	3333	1571	2535	3786	6678	2423	681	3125	3586	2636	817
21	3399	1484	578	3392	1511	888	4720	1588	3734	1096	887	1755	3600
22	1946	490	721	2013	908	1452	4074	209	2493	827	974	1608	2359
23	1278	990	1030	2922	324	1671	5071	1030	1934	908	2553	456	1876
24	669	1950	1990	2368	1116	2336	5724	1461	1377	1702	2624	1002	1267
25	627	2855	2895	2170	2183	3431	6675	2420	130	2773	3583	2245	200
26	2878	975	670	3336	1126	124	5000	1538	3402	587	1177	1246	3158
27	1737	926	835	3503	260	1165	4993	965	2449	844	1488	257	2339
28	3124	1214	909	3500	1367	189	5221	1716	3531	826	1398	1485	3397
29	2878	599	287	2884	996	640	4494	1112	3219	455	673	1225	3120
30	307	2535	2575	2109	1863	3110	6265	2010	343	2453	3173	1925	211
31	5217	3860	3803	5298	3961	4341	785	3482	5766	4041	3045	5166	5632
32	799	3027	3067	2108	2355	3603	6776	2521	228	2945	3684	2417	366
33	3305	1407	1102	3643	1565	489	5265	1839	3724	1019	1444	1678	3590
34	3716	1811	1505	4092	1959	855	5756	2288	4128	1423	1935	2082	4004
35	2251	359	158	2934	660	730	4628	1000	2883	75	941	871	2766
36	2878	1060	439	3093	1106	364	4747	1289	3277	572	926	1231	3134
37	3467	1557	1248	3843	1710	598	5511	2039	3870	1165	1686	1824	3736
38	4316	2959	2902	4391	3060	3440	1686	2551	4835	3140	2149	4265	4701
39	2963	394	287	2763	991	841	4276	955	3214	560	455	1290	3185
40	512	2740	2780	2083	2068	3316	6472	2217	184	2658	3380	2130	135
41	2515	98	276	2543	523	919	4447	715	2746	227	894	1019	2914
42	4850	3538	3436	4925	3594	3974	2230	3125	4875	3674	2688	4799	5274
43	1937	856	771	3236	434	1063	5167	1145	2492	502	1433	279	2267
44	367	2026	2066	2107	1354	2620	5784	1529	925	1944	2692	1441	815

45	3601	1710	1395	3939	1861	940	5561	2131	4017	1312	1740	1971	3883
46	3936	1733	1740	4274	2196	1090	5891	2466	4362	1650	2070	2309	4221
47	2430	508	407	3008	658	722	4902	1087	2985	170	1081	772	2667
48	2691	194	279	2522	750	921	4353	718	2973	383	696	1148	3043
49	2087	532	334	3006	563	761	4922	1083	2831	169	1101	618	2606
50	1861	2906	3135	806	2929	3866	6480	2225	1544	3136	3388	2844	1451
51	2358	435	337	2936	586	652	4759	1015	2913	98	1009	700	2595
52	2263	335	235	3026	668	754	4771	960	2891	80	950	802	2697
53	1425	2470	2699	947	2493	3430	6044	1789	1103	2700	2952	2413	1011
54	2266	137	451	2464	494	1096	4299	537	2717	411	1063	1188	2563
55	2166	234	546	2364	557	1191	4199	436	2655	508	964	1248	2623
56	3870	2072	1882	4208	2130	1232	5830	2400	4504	1584	2009	2243	4155
57	1417	1196	2699	3116	524	1582	5271	1238	1972	948	1972	248	1834
58	739	1517	1557	2390	851	2098	5404	1199	1447	1435	2286	1210	1337

VECTOR													
19	**33**	**56**	**34**	**45**	**15**	**14**	**28**	**39**	**10**	**43**	**57**	**12**	**30**
DISTANCIAS													
531	**413**	**300**	**354**	**562**	**665**	**224**	**959**	**560**	**502**	**527**	**248**	**1925**	**307**
SOLUCIÓN													
47918													

14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
3334	3987	2870	2601	330	3049	1302	3399	1946	1278	669	627	2878	1737	3124
1424	2185	1193	846	2142	1127	3104	1484	490	990	1950	2855	975	926	1214
1119	1776	1451	5410	2182	846	3333	578	721	1030	1990	2895	670	835	909
3710	4363	2941	3207	1991	3512	1571	3392	2013	2922	2368	2170	3336	3503	3500
1577	2230	1609	1003	1260	1297	2535	1511	908	324	1116	2183	1126	260	1367
347	1126	1989	250	2736	138	3786	888	1452	1671	2336	3431	124	1165	189
5423	6089	3142	4876	5676	5181	6678	4720	4074	5071	5724	6675	5000	4993	5221
1906	2559	1133	1352	1422	1651	2423	1588	209	1030	1461	2420	1538	965	1716
3741	4399	3417	3163	924	3462	681	3734	2493	1934	1377	130	3402	2449	3531
1036	1694	1694	458	1844	757	3125	1096	827	908	1702	2773	587	844	826
1609	2266	693	1041	2585	1353	3586	887	974	2553	2624	3583	1177	1488	1398
1695	2353	2819	1117	1557	1416	2636	1755	1608	456	1002	2245	1246	257	1485
3607	4265	3283	3029	931	3328	817	3600	2359	1876	1267	200	3158	2339	3397
0	665	2300	578	2845	474	4116	1216	1705	1901	2693	3848	652	1480	224
665	0	2957	1236	3706	992	4774	1471	2358	2654	3346	4589	1177	2151	887
2300	2957	0	1734	2543	2044	3544	1578	932	1939	2582	3304	1868	1861	2089
578	1236	1734	0	2267	300	3542	638	1202	1328	2094	3187	129	915	368
2845	3706	2543	2267	0	2566	1273	2905	1619	1175	564	1068	2600	1638	2841
474	992	2044	300	2566	0	3841	945	1507	1627	2393	3486	178	1214	254
4116	4774	3544	3542	1273	3841	0	3998	2620	2217	1647	693	3665	2676	3906
1216	1471	1578	638	2905	945	3998	0	1383	1835	2627	3846	767	1771	788
1705	2358	932	1202	1619	1507	2620	1383	0	1238	1612	2617	1332	1174	1495
1901	2654	1939	1328	1175	1627	2217	1835	1238	0	801	1865	1450	554	2243
2693	3346	2582	2094	564	2393	1647	2627	1612	801	0	1256	2083	1252	2324
3848	4589	3304	3187	1068	3486	693	3846	2617	1865	1256	0	3310	2324	3551
652	1177	1868	129	2600	178	3665	767	1332	1450	2083	3310	0	1044	432
1480	2151	1861	915	1638	1214	2676	1771	1174	554	1252	2324	1044	0	1275
224	887	2089	368	2841	254	3906	788	1495	2243	2324	3551	432	1275	0
968	1564	1362	390	2470	652	3487	520	875	1546	2058	3185	519	1023	758
3547	4099	3131	2866	770	3165	1052	3380	2211	1545	936	372	2995	2004	3236
4648	5304	2357	4091	5113	4396	5893	3925	3289	4286	4929	5890	4215	4208	4436
4020	4591	3642	3359	1240	3658	537	3872	2718	2037	1428	190	3487	2496	4225
176	818	2133	561	3039	531	4108	938	1634	1889	2517	3754	709	1468	283
394	262	2624	965	3433	730	4503	1200	2087	2293	2686	4148	908	1637	606
1153	1716	1491	480	2134	1079	3201	842	774	984	1841	2849	609	669	848
692	1350	1615	114	2580	374	3635	524	1109	1542	2057	3295	243	1022	482
137	528	2379	711	3184	476	4249	943	1838	2040	2806	3899	639	1627	359
3752	4408	1456	3190	4182	3495	4962	3024	2388	3385	4038	4959	3314	3307	3535
1169	1827	1144	591	2465	856	3366	607	754	1315	2123	3180	720	1088	959
3744	4306	3338	3072	953	3371	800	3585	2414	1750	1141	125	3200	2209	3441
1301	1905	1310	669	1997	968	3064	892	607	847	1849	2712	798	783	1037
4291	4947	1990	3724	4756	4029	5536	3558	2922	3919	4572	5533	3848	3841	4069
1420	2073	2035	846	1834	1140	2804	1470	1348	600	1275	2524	969	184	1210
3047	3590	2650	2379	114	2678	1385	2871	1726	1061	452	954	2486	1524	2727

479	562	2429	854	3335	806	4401	1288	1930	2182	2805	4050	984	1756	701
629	607	2759	1192	3670	960	4739	1438	2265	2510	3143	4385	1138	2094	851
1059	1717	1770	392	2388	691	3199	1118	991	1093	1610	2836	528	570	702
1249	2093	1216	671	2349	978	3129	742	517	1074	2020	2939	800	946	1039
1079	1742	1790	511	1984	810	3032	1143	983	887	1612	2644	640	523	879
4119	4772	3346	3565	1715	3864	1070	3801	2422	2425	1869	1669	3751	3178	3929
982	1645	1622	408	2068	707	3122	1046	919	1021	1542	2764	536	498	777
967	1747	1639	388	2074	687	3209	976	880	992	1642	2857	517	600	756
3683	4036	2910	3129	1284	3428	634	3365	1986	1994	1422	1233	3315	2742	3493
1424	2082	1162	846	1968	1145	2948	1076	459	818	1748	2683	975	754	1214
1421	2179	1062	950	2031	1249	2846	1174	359	881	1811	2746	1078	817	1311
771	760	2698	1126	3604	1096	4673	1185	2199	2454	3082	4319	1274	2033	1494
1939	2592	2139	1347	1340	1646	2395	1989	1438	432	754	2004	1329	498	1570
2901	3554	2278	1860	636	2159	1681	2951	1350	1394	259	1326	1189	1460	2532

1	**32**	**25**	**40**	**13**	**9**	**38**	**42**	**7**	**31**	**16**	**41**	**22**	**36**	**11**
799	**190**	**125**	**135**	**147**	**4835**	**544**	**2230**	**785**	**2357**	**1310**	**607**	**1109**	**926**	**1686**

29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
2878	307	5217	799	3305	3716	2251	2878	3467	4316	2963	512	2515	4850	1937
599	2535	3860	3027	1407	1811	359	1060	1557	2959	394	2740	98	3538	856
287	2575	3803	3067	1102	1505	158	439	1248	2902	287	2780	276	3436	771
2884	2109	5298	2108	3643	4092	2934	3093	3843	4391	2763	2083	2543	4925	3236
996	1863	3961	2355	1565	1959	660	1106	1710	3060	991	2068	523	3594	434
640	3110	4341	3603	489	855	730	364	598	3440	841	3316	919	3974	1063
4494	6265	785	6776	5265	5756	4628	4747	5511	1686	4276	6472	4447	2230	5167
1112	2010	3482	2521	1839	2288	1000	1289	2039	2551	955	2217	715	3125	1145
3219	343	5766	228	3724	4128	2883	3277	3870	4835	3214	184	2746	4875	2492
455	2453	4041	2945	1019	1423	75	572	1165	3140	560	2658	227	3674	502
673	3173	3045	3684	1444	1935	941	926	1686	2149	455	3380	894	2688	1433
1225	1925	5166	2417	1678	2082	871	1231	1824	4265	1290	2130	1019	4799	279
3120	211	5632	366	3590	4004	2766	3134	3736	4701	3185	135	2914	5274	2267
968	3547	4648	4020	176	394	1153	692	137	3752	1169	3744	1301	4291	1420
1564	4099	5304	4591	818	262	1716	1350	528	4408	1827	4306	1905	4947	2073
1362	3131	2357	3642	2133	2624	1491	1615	2379	1456	1144	3338	1310	1990	2035
390	2866	4091	3359	561	965	480	114	711	3190	591	3072	669	3724	846
2470	770	5113	1240	3039	3433	2134	2580	3184	4182	2465	953	1997	4756	1834
652	3165	4396	3658	531	730	1079	374	476	3495	856	3371	968	4029	1140
3487	1052	5893	537	4108	4503	3201	3635	4249	4962	3366	800	3064	5536	2804
520	3380	3925	3872	938	1200	842	524	943	3024	607	3585	892	3558	1470
875	2211	3289	2718	1634	2087	774	1109	1838	2388	754	2414	607	2922	1348
1546	1545	4286	2037	1889	2293	984	1542	2040	3385	1315	1750	847	3919	600
2058	936	4929	1428	2517	2686	1841	2057	2806	4038	2123	1141	1849	4572	1275
3185	372	5890	190	3754	4148	2849	3295	3899	4959	3180	125	2712	5533	2524
519	2995	4215	3487	709	908	609	243	639	3314	720	3200	798	3848	969
1023	2004	4208	2496	1468	1637	669	1022	1627	3307	1088	2209	783	3841	184
758	3236	4436	4225	283	606	848	482	359	3535	959	3441	1037	4069	1210
0	2865	3709	3036	478	1293	354	274	1036	2808	366	3070	447	3342	949
2865	0	5480	549	3434	3828	2529	2975	3579	4549	2860	257	2392	5123	2204
3709	5480	0	5991	4480	4971	3843	3962	4791	901	3491	5687	3662	1445	4615
3036	549	5991	0	3926	4320	3021	3467	4171	5060	3352	297	2884	5634	2696
478	3434	4480	3926	0	547	1041	675	290	3579	1152	3639	1230	4113	1402
1293	3828	4971	4320	547	0	1444	1016	257	4002	1493	4041	1638	4604	1802
354	2529	3843	3021	1041	1444	0	594	1187	2942	329	2734	148	3476	567
274	2975	3962	3467	675	1016	594	0	759	3061	705	3180	635	3595	971
1036	3579	4791	4171	290	257	1187	759	0	3825	1236	3784	1381	4359	1553
2808	4549	901	5060	3579	4002	2942	3061	3825	0	2590	4848	2761	544	3481
366	2860	3491	3352	1152	1493	329	705	1236	2590	0	3065	432	3124	1014
3070	257	5687	297	3639	4041	2734	3180	3784	4848	3065	0	2597	5330	2409
447	2392	3662	2884	1230	1638	148	635	1381	2761	432	2597	0	3294	722
3342	5123	1445	5634	4113	4604	3476	3595	4359	544	3124	5330	3294	0	4015
949	2204	4615	2696	1402	1802	567	971	1553	3481	1014	2409	722	4015	0
2356	656	4999	1126	2925	3319	2020	2466	3070	4068	2351	839	1883	4642	1720

708	3730	4776	4222	213	354	1335	968	342	3875	1445	3935	1423	4409	1698
837	4065	5106	4557	435	229	1672	1306	492	4505	1783	4270	1783	4739	2037
577	2527	4117	3008	802	1356	243	506	1099	3145	602	2721	391	3750	554
367	2619	3568	3114	1232	1849	287	705	1592	2667	345	2923	192	3201	852
623	2354	4137	2846	1072	1475	268	625	1218	3236	688	2559	416	3770	339
3325	1608	5695	1607	4052	4501	3213	3502	4252	4764	3168	1582	2928	5338	3004
525	2444	3969	2936	970	1139	171	524	1115	3073	590	2649	319	3607	429
523	2537	3986	3162	949	1352	107	502	1095	3083	488	2753	217	3617	533
2889	1172	5259	1171	3616	4065	2777	3066	3816	4328	2732	1146	2492	4902	2571
631	2406	3514	2855	1407	1839	315	960	1560	2613	498	2568	184	3147	889
728	2305	3414	2918	1511	1931	419	1064	1672	2513	589	2651	282	3047	993
1635	3999	5045	4491	413	300	1606	1240	634	4144	1717	4204	1795	4678	1967
1304	1684	4708	2176	1763	1932	1087	1303	1908	3590	1369	1889	1095	4124	527
2266	1006	4619	1498	2725	2899	2049	1974	2570	3718	2331	1211	2042	4252	1123

37	**46**	**5**	**23**	**29**	**17**	**52**	**47**	**35**	**51**	**26**	**48**	**49**	**21**	**3**
492	**2196**	**324**	**1546**	**390**	**388**	**170**	**243**	**171**	**536**	**800**	**543**	**1143**	**578**	**314**

44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
367	3601	3936	2430	2691	2087	1861	2358	2263	1425	2266	2166	3870	1417	739
2026	1710	1733	508	194	532	2906	435	335	2470	137	234	2072	1196	1517
2066	1395	1740	407	279	334	3135	337	235	2699	451	546	1882	2699	1557
2107	3939	4274	3008	2522	3006	806	2936	3026	947	2464	2364	4208	3116	2390
1354	1861	2196	658	750	563	2929	586	668	2493	494	557	2130	524	851
2620	940	1090	722	921	761	3866	652	754	3430	1096	1191	1232	1582	2098
5784	5561	5891	4902	4353	4922	6480	4759	4771	6044	4299	4199	5830	5271	5404
1529	2131	2466	1087	718	1083	2225	1015	960	1789	537	436	2400	1238	1199
925	4017	4362	2985	2973	2831	1544	2913	2891	1103	2717	2655	4504	1972	1447
1944	1312	1650	170	383	169	3136	98	80	2700	411	508	1584	948	1435
2692	1740	2070	1081	696	1101	3388	1009	950	2952	1063	964	2009	1972	2286
1441	1971	2309	772	1148	618	2844	700	802	2413	1188	1248	2243	248	1210
815	3883	4221	2667	3043	2606	1451	2595	2697	1011	2563	2623	4155	1834	1337
3047	479	629	1059	1249	1079	4119	982	967	3683	1424	1421	771	1939	2901
3590	562	607	1717	2093	1742	4772	1645	1747	4036	2082	2179	760	2592	3554
2650	2429	2759	1770	1216	1790	3346	1622	1639	2910	1162	1062	2698	2139	2278
2379	854	1192	392	671	511	3565	408	388	3129	846	950	1126	1347	1860
114	3335	3670	2388	2349	1984	1715	2068	2074	1284	1968	2031	3604	1340	636
2678	806	960	691	978	810	3864	707	687	3428	1145	1249	1096	1646	2159
1385	4401	4739	3199	3129	3032	1070	3122	3209	634	2948	2846	4673	2395	1681
2871	1288	1438	1118	742	1143	3801	1046	976	3365	1076	1174	1185	1989	2951
1726	1930	2265	991	517	983	2422	919	880	1986	459	359	2199	1438	1350
1061	2182	2510	1093	1074	887	2425	1021	992	1994	818	881	2454	432	1394
452	2805	3143	1610	2020	1612	1869	1542	1642	1422	1748	1811	3082	754	259
954	4050	4385	2836	2939	2644	1669	2764	2857	1233	2683	2746	4319	2004	1326
2486	984	1138	528	800	640	3751	536	517	3315	975	1078	1274	1329	1189
1524	1756	2094	570	946	523	3178	498	600	2742	754	817	2033	498	1460
2727	701	851	702	1039	879	3929	777	756	3493	1214	1311	1494	1570	2532
2356	708	837	577	367	623	3325	525	523	2889	631	728	1635	1304	2266
656	3730	4065	2527	2619	2354	1608	2444	2537	1172	2406	2305	3999	1684	1006
4999	4776	5106	4117	3568	4137	5695	3969	3986	5259	3514	3414	5045	4708	4619
1126	4222	4557	3008	3114	2846	1607	2936	3162	1171	2855	2918	4491	2176	1498
2925	213	435	802	1232	1072	4052	970	949	3616	1407	1511	413	1763	2725
3319	354	229	1356	1849	1475	4501	1139	1352	4065	1839	1931	300	1932	2899
2020	1335	1672	243	287	268	3213	171	107	2777	315	419	1606	1087	2049
2466	968	1306	506	705	625	3502	524	502	3066	960	1064	1240	1303	1974
3070	342	492	1099	1592	1218	4252	1115	1095	3816	1560	1672	634	1908	2570
4068	3875	4505	3145	2667	3236	4764	3073	3083	4328	2613	2513	4144	3590	3718
2351	1445	1783	602	345	688	3168	590	488	2732	498	589	1717	1369	2331
839	3935	4270	2721	2923	2559	1582	2649	2753	1146	2568	2651	4204	1889	1211
1883	1423	1783	391	192	416	2928	319	217	2492	184	282	1795	1095	2042
4642	4409	4739	3750	3201	3770	5338	3607	3617	4902	3147	3047	4678	4124	4252
1720	1698	2037	554	852	339	3004	429	533	2571	889	993	1967	527	1123
0	3221	3556	2274	2235	1870	1601	1954	1960	1170	1854	1917	3490	1226	522

3221	0	273	1340	1525	1355	4344	1258	942	3908	1700	1798	312	2051	3033
3556	273	0	1668	1863	1693	4679	1596	1580	4243	2038	2134	254	2389	3351
2274	1340	1668	0	520	86	3299	72	170	2864	558	662	1489	856	1434
2235	1525	1863	520	0	543	2931	448	342	2495	239	288	1797	1266	1878
1870	1355	1693	86	543	0	3295	97	199	2859	588	692	1637	858	1339
1601	4344	4679	3299	2931	3295	0	3227	3206	446	2750	2649	4617	2417	1882
1954	1258	1596	72	448	97	3227	0	102	2791	486	590	1535	8700	1441
1960	942	1580	170	342	199	3206	102	0	2770	389	497	1514	888	1543
1170	3908	4243	2864	2495	2859	446	2791	2770	0	2314	2213	4181	2105	1446
1854	1700	2038	558	239	588	2750	486	389	2314	0	112	1972	994	1345
1917	1798	2134	662	288	692	2649	590	497	2213	112	0	2076	1057	1408
3490	312	254	1489	1797	1637	4617	1535	1514	4181	1972	2076	0	2328	2986
1226	2051	2389	856	1266	858	2417	8700	888	2105	994	1057	2328	0	962
522	3033	3351	1434	1878	1339	1882	1441	1543	1446	1345	1408	2986	962	0

2	**54**	**55**	**8**	**4**	**50**	**53**	**18**	**24**	**20**	**44**	**58**	**27**	**6**	**19**

137	**112**	**436**	**2120**	**806**	**446**	**1284**	**564**	**1647**	**1385**	**522**	**1460**	**1165**	**138**

SOLUCIÓN DEL BRG180 USANDO INDICE Y SOLVER

180

	1	2	3	4	5	6
1	0	20	10000	10000	10000	10000
2	20	0	0	10000	10000	10000
3	10000	0	0	20	10000	10000
4	10000	10000	20	0	0	10000
5	10000	10000	10000	0	0	20
6	10000	10000	10000	10000	20	0
7	10000	10000	10000	10000	10000	0
8	10000	10000	10000	10000	10000	10000
9	10000	10000	10000	10000	10000	10000
10	10000	10000	10000	10000	10000	10000
11	10000	10000	10000	10000	10000	10000
12	0	10000	10000	10000	10000	10000
13	30	9000	30	3500	30	3500
14	30	9000	30	3500	30	3500
15	30	3500	30	9000	30	3500
16	30	3500	30	9000	30	3500
17	30	3500	30	3500	30	9000
18	30	3500	30	3500	30	9000
19	30	3500	30	3500	30	9000
20	30	3500	30	3500	30	9000
21	30	3500	30	9000	30	3500
22	30	3500	30	9000	30	3500
23	30	9000	30	3500	30	3500
24	30	9000	30	3500	30	3500
25	30	9000	30	3500	30	3500
26	30	9000	30	3500	30	3500
27	30	3500	30	3500	30	9000
28	30	3500	30	3500	30	9000
29	30	3500	30	9000	30	3500
30	30	3500	30	9000	30	3500
31	30	3500	30	9000	30	3500
32	30	3500	30	9000	30	3500
33	30	3500	30	3500	30	9000
34	30	3500	30	3500	30	9000
35	30	9000	30	3500	30	3500
36	30	9000	30	3500	30	3500
37	30	3500	30	9000	30	3500
38	30	3500	30	9000	30	3500
39	30	3500	30	3500	30	9000
40	30	3500	30	3500	30	9000
41	30	9000	30	3500	30	3500
42	30	9000	30	3500	30	3500
43	30	9000	30	3500	30	3500
44	30	9000	30	3500	30	3500

45	30	3500	30	3500	30	9000
46	30	3500	30	3500	30	9000
47	30	3500	30	9000	30	3500
48	30	3500	30	9000	30	3500
49	30	3500	30	3500	30	9000
50	30	3500	30	3500	30	9000
51	30	3500	30	9000	30	3500
52	30	3500	30	9000	30	3500
53	30	9000	30	3500	30	3500
54	30	9000	30	3500	30	3500
55	30	9000	30	3500	30	3500
56	30	9000	30	3500	30	3500
57	30	3500	30	9000	30	3500
58	30	3500	30	9000	30	3500
59	30	3500	30	3500	30	9000
60	30	3500	30	3500	30	9000
61	30	9000	30	3500	30	3500
62	30	9000	30	3500	30	3500
63	30	3500	30	9000	30	3500
64	30	3500	30	9000	30	3500
65	30	3500	30	3500	30	9000
66	30	3500	30	3500	30	9000
67	30	3500	30	3500	30	9000
68	30	3500	30	3500	30	9000
69	30	3500	30	9000	30	3500
70	30	3500	30	9000	30	3500
71	30	9000	30	3500	30	3500
72	30	9000	30	3500	30	3500
73	30	9000	30	3500	30	3500
74	30	9000	30	3500	30	3500
75	30	3500	30	3500	30	9000
76	30	3500	30	3500	30	9000
77	30	3500	30	9000	30	3500
78	30	3500	30	9000	30	3500
79	30	3500	30	9000	30	3500
80	30	3500	30	9000	30	3500
81	30	3500	30	3500	30	9000
82	30	3500	30	3500	30	9000
83	30	9000	30	3500	30	3500
84	30	9000	30	3500	30	3500
85	30	3500	30	9000	30	3500
86	30	3500	30	9000	30	3500
87	30	3500	30	3500	30	9000
88	30	3500	30	3500	30	9000
89	30	9000	30	3500	30	3500
90	30	9000	30	3500	30	3500
91	30	9000	30	3500	30	3500
92	30	9000	30	3500	30	3500

93	30	3500	30	3500	30	9000
94	30	3500	30	3500	30	9000
95	30	3500	30	9000	30	3500
96	30	3500	30	9000	30	3500
97	30	3500	30	3500	30	9000
98	30	3500	30	3500	30	9000
99	30	3500	30	9000	30	3500
100	30	3500	30	9000	30	3500
101	30	9000	30	3500	30	3500
102	30	9000	30	3500	30	3500
103	30	9000	30	3500	30	3500
104	30	9000	30	3500	30	3500
105	30	3500	30	9000	30	3500
106	30	3500	30	9000	30	3500
107	30	3500	30	3500	30	9000
108	30	3500	30	3500	30	9000
109	30	9000	30	9000	30	9000
110	30	9000	30	9000	30	9000
111	30	9000	30	30	30	30
112	30	9000	30	30	30	30
113	30	9000	30	30	30	30
114	30	9000	30	30	30	30
115	30	9000	30	30	30	30
116	30	9000	30	30	30	30
117	30	9000	30	30	30	30
118	30	9000	30	30	30	30
119	30	9000	30	9000	30	9000
120	30	9000	30	9000	30	9000
121	30	9000	30	9000	30	9000
122	30	9000	30	9000	30	9000
123	30	30	30	9000	30	30
124	30	30	30	9000	30	30
125	30	30	30	9000	30	30
126	30	30	30	9000	30	30
127	30	30	30	9000	30	30
128	30	30	30	9000	30	30
129	30	30	30	9000	30	30
130	30	30	30	9000	30	30
131	30	9000	30	9000	30	9000
132	30	9000	30	9000	30	9000
133	30	9000	30	9000	30	9000
134	30	9000	30	9000	30	9000
135	30	30	30	30	30	9000
136	30	30	30	30	30	9000
137	30	30	30	30	30	9000
138	30	30	30	30	30	9000
139	30	30	30	30	30	9000
140	30	30	30	30	30	9000

141	30	30	30	30	30	9000
142	30	30	30	30	30	9000
143	30	9000	30	9000	30	9000
144	30	9000	30	9000	30	9000
145	30	9000	30	9000	30	9000
146	30	9000	30	9000	30	9000
147	30	30	30	30	30	9000
148	30	30	30	30	30	9000
149	30	30	30	30	30	9000
150	30	30	30	30	30	9000
151	30	30	30	30	30	9000
152	30	30	30	30	30	9000
153	30	30	30	30	30	9000
154	30	30	30	30	30	9000
155	30	9000	30	9000	30	9000
156	30	9000	30	9000	30	9000
157	30	9000	30	9000	30	9000
158	30	9000	30	9000	30	9000
159	30	30	30	9000	30	30
160	30	30	30	9000	30	30
161	30	30	30	9000	30	30
162	30	30	30	9000	30	30
163	30	30	30	9000	30	30
164	30	30	30	9000	30	30
165	30	30	30	9000	30	30
166	30	30	30	9000	30	30
167	30	9000	30	9000	30	9000
168	30	9000	30	9000	30	9000
169	30	9000	30	9000	30	9000
170	30	9000	30	9000	30	9000
171	30	9000	30	30	30	30
172	30	9000	30	30	30	30
173	30	9000	30	30	30	30
174	30	9000	30	30	30	30
175	30	9000	30	30	30	30
176	30	9000	30	30	30	30
177	30	9000	30	30	30	30
178	30	9000	30	30	30	30
179	30	9000	30	9000	30	9000
180	30	9000	30	9000	30	9000

VECTOR						
1	**2**	**3**	**4**	**5**	**6**	**7**
DISTANCIAS						
20	**0**	**20**	**0**	**20**	**0**	**20**
SOLUCIÓN						
118860						

7	8	9	10	11	12	13	14
10000	10000	10000	10000	10000	0	30	30
10000	10000	10000	10000	10000	10000	9000	9000
10000	10000	10000	10000	10000	10000	30	30
10000	10000	10000	10000	10000	10000	3500	3500
10000	10000	10000	10000	10000	10000	30	30
0	10000	10000	10000	10000	10000	3500	3500
0	20	10000	10000	10000	10000	30	30
20	0	0	10000	10000	10000	3500	3500
10000	0	0	20	10000	10000	30	30
10000	10000	20	0	0	10000	3500	3500
10000	10000	10000	0	0	20	30	30
10000	10000	10000	10000	20	0	9000	9000
30	3500	30	3500	30	9000	0	20
30	3500	30	3500	30	9000	20	0
30	3500	30	9000	30	3500	10000	0
30	3500	30	9000	30	3500	10000	10000
30	9000	30	3500	30	3500	10000	10000
30	9000	30	3500	30	3500	10000	10000
30	9000	30	3500	30	3500	10000	10000
30	9000	30	3500	30	3500	10000	10000
30	3500	30	9000	30	3500	10000	10000
30	3500	30	9000	30	3500	10000	10000
30	3500	30	3500	30	9000	10000	10000
30	3500	30	3500	30	9000	0	10000
30	3500	30	3500	30	9000	9000	9000
30	3500	30	3500	30	9000	9000	9000
30	9000	30	3500	30	3500	3500	3500
30	9000	30	3500	30	3500	3500	3500
30	3500	30	9000	30	3500	3500	3500
30	3500	30	9000	30	3500	3500	3500
30	3500	30	9000	30	3500	3500	3500
30	3500	30	9000	30	3500	3500	3500
30	9000	30	3500	30	3500	3500	3500
30	9000	30	3500	30	3500	3500	3500
30	3500	30	3500	30	9000	9000	9000
30	3500	30	3500	30	9000	9000	9000
30	3500	30	9000	30	3500	3500	3500
30	3500	30	9000	30	3500	3500	3500
30	9000	30	3500	30	3500	9000	9000
30	9000	30	3500	30	3500	9000	9000
30	3500	30	3500	30	9000	3500	3500
30	3500	30	3500	30	9000	3500	3500
30	3500	30	3500	30	9000	3500	3500
30	3500	30	3500	30	9000	3500	3500

30	9000	30	3500	30	3500	9000	9000
30	9000	30	3500	30	3500	9000	9000
30	3500	30	9000	30	3500	3500	3500
30	3500	30	9000	30	3500	3500	3500
30	9000	30	3500	30	3500	3500	3500
30	9000	30	3500	30	3500	3500	3500
30	3500	30	9000	30	3500	9000	9000
30	3500	30	9000	30	3500	9000	9000
30	3500	30	3500	30	9000	3500	3500
30	3500	30	3500	30	9000	3500	3500
30	3500	30	3500	30	9000	3500	3500
30	3500	30	3500	30	9000	3500	3500
30	3500	30	9000	30	3500	9000	9000
30	3500	30	9000	30	3500	9000	9000
30	9000	30	3500	30	3500	3500	3500
30	9000	30	3500	30	3500	3500	3500
30	3500	30	3500	30	9000	9000	9000
30	3500	30	3500	30	9000	9000	9000
30	3500	30	9000	30	3500	3500	3500
30	3500	30	9000	30	3500	3500	3500
30	9000	30	3500	30	3500	3500	3500
30	9000	30	3500	30	3500	3500	3500
30	9000	30	3500	30	3500	3500	3500
30	9000	30	3500	30	3500	3500	3500
30	3500	30	9000	30	3500	3500	3500
30	3500	30	9000	30	3500	3500	3500
30	3500	30	3500	30	9000	9000	9000
30	3500	30	3500	30	9000	9000	9000
30	3500	30	3500	30	9000	9000	9000
30	3500	30	3500	30	9000	9000	9000
30	9000	30	3500	30	3500	9000	9000
30	9000	30	3500	30	3500	9000	9000
30	3500	30	9000	30	3500	9000	9000
30	3500	30	9000	30	3500	9000	9000
30	3500	30	9000	30	3500	9000	9000
30	3500	30	9000	30	3500	9000	9000
30	9000	30	3500	30	3500	9000	9000
30	9000	30	3500	30	3500	9000	9000
30	3500	30	3500	30	9000	9000	9000
30	3500	30	3500	30	9000	9000	9000
30	3500	30	9000	30	3500	9000	9000
30	3500	30	9000	30	3500	9000	9000
30	9000	30	3500	30	3500	30	30
30	9000	30	3500	30	3500	30	30
30	3500	30	3500	30	9000	30	30
30	3500	30	3500	30	9000	30	30
30	3500	30	3500	30	9000	30	30
30	3500	30	3500	30	9000	30	30

30	9000	30	3500	30	3500	30	30
30	9000	30	3500	30	3500	30	30
30	3500	30	9000	30	3500	9000	9000
30	3500	30	9000	30	3500	9000	9000
30	9000	30	3500	30	3500	9000	9000
30	9000	30	3500	30	3500	9000	9000
30	3500	30	9000	30	3500	30	30
30	3500	30	9000	30	3500	30	30
30	3500	30	3500	30	9000	30	30
30	3500	30	3500	30	9000	30	30
30	3500	30	3500	30	9000	30	30
30	3500	30	3500	30	9000	30	30
30	3500	30	9000	30	3500	30	30
30	3500	30	9000	30	3500	30	30
30	9000	30	3500	30	3500	9000	9000
30	9000	30	3500	30	3500	9000	9000
30	9000	30	9000	30	9000	9000	9000
30	9000	30	9000	30	9000	9000	9000
30	30	30	30	30	9000	3500	3500
30	30	30	30	30	9000	3500	3500
30	30	30	30	30	9000	3500	3500
30	30	30	30	30	9000	3500	3500
30	30	30	30	30	9000	3500	3500
30	30	30	30	30	9000	3500	3500
30	30	30	30	30	9000	3500	3500
30	30	30	30	30	9000	3500	3500
30	9000	30	9000	30	9000	9000	9000
30	9000	30	9000	30	9000	9000	9000
30	9000	30	9000	30	9000	3500	3500
30	9000	30	9000	30	9000	3500	3500
30	30	30	9000	30	30	3500	3500
30	30	30	9000	30	30	3500	3500
30	30	30	9000	30	30	9000	9000
30	30	30	9000	30	30	9000	9000
30	30	30	9000	30	30	9000	9000
30	30	30	9000	30	30	9000	9000
30	30	30	9000	30	30	3500	3500
30	30	30	9000	30	30	3500	3500
30	9000	30	9000	30	9000	3500	3500
30	9000	30	9000	30	9000	3500	3500
30	9000	30	9000	30	9000	3500	3500
30	9000	30	9000	30	9000	3500	3500
30	9000	30	30	30	30	3500	3500
30	9000	30	30	30	30	3500	3500
30	9000	30	30	30	30	9000	9000
30	9000	30	30	30	30	9000	9000
30	9000	30	30	30	30	9000	9000
30	9000	30	30	30	30	9000	9000

30	9000	30	30	30	30	3500	3500
30	9000	30	30	30	30	3500	3500
30	9000	30	9000	30	9000	3500	3500
30	9000	30	9000	30	9000	3500	3500
30	9000	30	9000	30	9000	30	30
30	9000	30	9000	30	9000	30	30
30	9000	30	30	30	30	9000	9000
30	9000	30	30	30	30	9000	9000
30	9000	30	30	30	30	30	30
30	9000	30	30	30	30	30	30
30	9000	30	30	30	30	30	30
30	9000	30	30	30	30	30	30
30	9000	30	30	30	30	9000	9000
30	9000	30	30	30	30	9000	9000
30	9000	30	9000	30	9000	30	30
30	9000	30	9000	30	9000	30	30
30	9000	30	9000	30	9000	30	30
30	9000	30	9000	30	9000	30	30
30	30	30	9000	30	30	9000	9000
30	30	30	9000	30	30	9000	9000
30	30	30	9000	30	30	30	30
30	30	30	9000	30	30	30	30
30	30	30	9000	30	30	30	30
30	30	30	9000	30	30	30	30
30	30	30	9000	30	30	9000	9000
30	30	30	9000	30	30	9000	9000
30	9000	30	9000	30	9000	30	30
30	9000	30	9000	30	9000	30	30
30	9000	30	9000	30	9000	9000	9000
30	9000	30	9000	30	9000	9000	9000
30	30	30	30	30	9000	9000	9000
30	30	30	30	30	9000	9000	9000
30	30	30	30	30	9000	9000	9000
30	30	30	30	30	9000	9000	9000
30	30	30	30	30	9000	9000	9000
30	30	30	30	30	9000	9000	9000
30	30	30	30	30	9000	9000	9000
30	30	30	30	30	9000	9000	9000
30	9000	30	9000	30	9000	9000	9000
30	9000	30	9000	30	9000	9000	9000
8	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	**20**	**0**	**20**	**9000**	**20**	**0**	**20**

15	16	17	18	19	20	21	22
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	3500	3500
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	9000	9000	9000	9000	3500	3500
30	30	30	30	30	30	30	30
3500	3500	9000	9000	9000	9000	3500	3500
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	3500	3500
10000	10000	10000	10000	10000	10000	10000	10000
0	10000	10000	10000	10000	10000	10000	10000
0	20	10000	10000	10000	10000	10000	10000
20	0	0	10000	10000	10000	10000	10000
10000	0	0	20	10000	10000	10000	10000
10000	10000	20	0	0	10000	10000	10000
10000	10000	10000	0	0	20	10000	10000
10000	10000	10000	10000	20	0	0	10000
10000	10000	10000	10000	10000	0	0	20
10000	10000	10000	10000	10000	10000	20	0
10000	10000	10000	10000	10000	10000	10000	0
10000	10000	10000	10000	10000	10000	10000	10000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500

3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000

9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500

9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
16	**17**	**18**	**19**	**20**	**21**	**22**	**23**
0	**20**	**0**	**20**	**0**	**20**	**0**	**20**

23	24	25	26	27	28	29	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	3500	3500	3500	3500
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	3500	3500
30	30	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	3500	3500
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
9000	9000	9000	9000	3500	3500	3500	3500
10000	0	9000	9000	3500	3500	3500	3500
10000	10000	9000	9000	3500	3500	3500	3500
10000	10000	3500	3500	9000	9000	3500	3500
10000	10000	3500	3500	9000	9000	3500	3500
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	9000	9000	3500	3500
0	10000	3500	3500	9000	9000	3500	3500
0	20	9000	9000	3500	3500	3500	3500
20	0	9000	9000	3500	3500	3500	3500
9000	9000	0	20	10000	10000	10000	10000
9000	9000	20	0	0	10000	10000	10000
3500	3500	10000	0	0	20	10000	10000
3500	3500	10000	10000	20	0	0	10000
3500	3500	10000	10000	10000	0	0	20
3500	3500	10000	10000	10000	10000	20	0
3500	3500	10000	10000	10000	10000	10000	0
3500	3500	10000	10000	10000	10000	10000	10000
3500	3500	10000	10000	10000	10000	10000	10000
3500	3500	10000	10000	10000	10000	10000	10000
9000	9000	10000	10000	10000	10000	10000	10000
9000	9000	0	10000	10000	10000	10000	10000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000

9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30

30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30

3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30

24	**25**	**26**	**27**	**28**	**29**	**30**	**31**
9000	**20**	**0**	**20**	**0**	**20**	**0**	**20**

31	32	33	34	35	36	37	38
30	30	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	3500	3500
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	9000	9000	3500	3500	3500	3500
30	30	30	30	30	30	30	30
3500	3500	9000	9000	3500	3500	3500	3500
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
10000	10000	10000	10000	10000	0	9000	9000
10000	10000	10000	10000	10000	10000	9000	9000
10000	10000	10000	10000	10000	10000	3500	3500
10000	10000	10000	10000	10000	10000	3500	3500
10000	10000	10000	10000	10000	10000	3500	3500
0	10000	10000	10000	10000	10000	3500	3500
0	20	10000	10000	10000	10000	3500	3500
20	0	0	10000	10000	10000	3500	3500
10000	0	0	20	10000	10000	3500	3500
10000	10000	20	0	0	10000	3500	3500
10000	10000	10000	0	0	20	9000	9000
10000	10000	10000	10000	20	0	9000	9000
3500	3500	3500	3500	9000	9000	0	20
3500	3500	3500	3500	9000	9000	20	0
3500	3500	9000	9000	3500	3500	10000	0
3500	3500	9000	9000	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000

3500	3500	9000	9000	3500	3500	10000	10000
3500	3500	9000	9000	3500	3500	10000	10000
3500	3500	3500	3500	9000	9000	10000	10000
3500	3500	3500	3500	9000	9000	0	10000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500

9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	9000	9000	30	30	30	30
30	30	9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000

9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
30	30	9000	9000	30	30	30	30
30	30	9000	9000	30	30	30	30
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
32	**33**	**34**	**35**	**36**	**37**	**38**	**39**
0	**20**	**0**	**20**	**9000**	**20**	**0**	**20**

39	40	41	42	43	44	45	46
30	30	30	30	30	30	30	30
3500	3500	9000	9000	9000	9000	3500	3500
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	3500	3500
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	3500	3500
30	30	30	30	30	30	30	30
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
10000	10000	10000	10000	10000	10000	10000	10000
0	10000	10000	10000	10000	10000	10000	10000
0	20	10000	10000	10000	10000	10000	10000
20	0	0	10000	10000	10000	10000	10000
10000	0	0	20	10000	10000	10000	10000
10000	10000	20	0	0	10000	10000	10000
10000	10000	10000	0	0	20	10000	10000
10000	10000	10000	10000	20	0	0	10000

10000	10000	10000	10000	10000	0	0	20
10000	10000	10000	10000	10000	10000	20	0
10000	10000	10000	10000	10000	10000	10000	0
10000	10000	10000	10000	10000	10000	10000	10000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500

9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000

9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500

40	**41**	**42**	**43**	**44**	**45**	**46**	**47**
0	**20**	**0**	**20**	**0**	**20**	**0**	**20**

47	48	49	50	51	52	53	54
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
9000	9000	3500	3500	9000	9000	3500	3500
30	30	30	30	30	30	30	30
3500	3500	9000	9000	3500	3500	3500	3500
30	30	30	30	30	30	30	30
3500	3500	9000	9000	3500	3500	3500	3500
30	30	30	30	30	30	30	30
9000	9000	3500	3500	9000	9000	3500	3500
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
10000	0	9000	9000	3500	3500	3500	3500
10000	10000	9000	9000	3500	3500	3500	3500
10000	10000	3500	3500	9000	9000	3500	3500
10000	10000	3500	3500	9000	9000	3500	3500
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000

10000	10000	3500	3500	9000	9000	3500	3500
0	10000	3500	3500	9000	9000	3500	3500
0	20	9000	9000	3500	3500	3500	3500
20	0	9000	9000	3500	3500	3500	3500
9000	9000	0	20	10000	10000	10000	10000
9000	9000	20	0	0	10000	10000	10000
3500	3500	10000	0	0	20	10000	10000
3500	3500	10000	10000	20	0	0	10000
3500	3500	10000	10000	10000	0	0	20
3500	3500	10000	10000	10000	10000	20	0
3500	3500	10000	10000	10000	10000	10000	0
3500	3500	10000	10000	10000	10000	10000	10000
3500	3500	10000	10000	10000	10000	10000	10000
3500	3500	10000	10000	10000	10000	10000	10000
9000	9000	10000	10000	10000	10000	10000	10000
9000	9000	0	10000	10000	10000	10000	10000
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000

3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500

30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
48	**49**	**50**	**51**	**52**	**53**	**54**	**55**
9000	**20**	**0**	**20**	**0**	**20**	**0**	**20**

55	56	57	58	59	60	61	62
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	9000	9000	3500	3500	3500	3500
30	30	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	3500	3500
30	30	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	3500	3500
30	30	30	30	30	30	30	30
3500	3500	9000	9000	3500	3500	3500	3500
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30

3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
10000	10000	10000	10000	10000	0	9000	9000
10000	10000	10000	10000	10000	10000	9000	9000
10000	10000	10000	10000	10000	10000	30	30
10000	10000	10000	10000	10000	10000	30	30
10000	10000	10000	10000	10000	10000	30	30
0	10000	10000	10000	10000	10000	30	30
0	20	10000	10000	10000	10000	30	30
20	0	0	10000	10000	10000	30	30
10000	0	0	20	10000	10000	30	30
10000	10000	20	0	0	10000	30	30
10000	10000	10000	0	0	20	9000	9000
10000	10000	10000	10000	20	0	9000	9000
30	30	30	30	9000	9000	0	20
30	30	30	30	9000	9000	20	0
30	30	30	30	9000	9000	10000	0
30	30	30	30	9000	9000	10000	10000
9000	9000	9000	9000	9000	9000	10000	10000
9000	9000	9000	9000	9000	9000	10000	10000
9000	9000	9000	9000	9000	9000	10000	10000
9000	9000	9000	9000	9000	9000	10000	10000
30	30	30	30	9000	9000	10000	10000
30	30	30	30	9000	9000	10000	10000
30	30	30	30	9000	9000	10000	10000
30	30	30	30	9000	9000	0	10000
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500

9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500

9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
56	**57**	**58**	**59**	**60**	**61**	**62**	**63**
0	**20**	**0**	**20**	**9000**	**20**	**0**	**20**

63	64	65	66	67	68	69	70
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	3500	3500
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	9000	9000	9000	9000	3500	3500
30	30	30	30	30	30	30	30
3500	3500	9000	9000	9000	9000	3500	3500
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000

9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
10000	10000	10000	10000	10000	10000	10000	10000
0	10000	10000	10000	10000	10000	10000	10000
0	20	10000	10000	10000	10000	10000	10000
20	0	0	10000	10000	10000	10000	10000
10000	0	0	20	10000	10000	10000	10000
10000	10000	20	0	0	10000	10000	10000
10000	10000	10000	0	0	20	10000	10000
10000	10000	10000	10000	20	0	0	10000
10000	10000	10000	10000	10000	0	0	20
10000	10000	10000	10000	10000	10000	20	0
10000	10000	10000	10000	10000	10000	10000	0
10000	10000	10000	10000	10000	10000	10000	10000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500

3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000

3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
64	**65**	**66**	**67**	**68**	**69**	**70**	**71**
0	**20**	**0**	**20**	**0**	**20**	**0**	**20**

71	72	73	74	75	76	77	78
30	30	30	30	30	30	30	30
9000	9000	9000	9000	3500	3500	3500	3500
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	3500	3500
30	30	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	3500	3500
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30

30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
10000	0	9000	9000	3500	3500	3500	3500
10000	10000	9000	9000	3500	3500	3500	3500
10000	10000	3500	3500	9000	9000	3500	3500
10000	10000	3500	3500	9000	9000	3500	3500
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	9000	9000	3500	3500
0	10000	3500	3500	9000	9000	3500	3500
0	20	9000	9000	3500	3500	3500	3500
20	0	9000	9000	3500	3500	3500	3500
9000	9000	0	20	10000	10000	10000	10000
9000	9000	20	0	0	10000	10000	10000
3500	3500	10000	0	0	20	10000	10000
3500	3500	10000	10000	20	0	0	10000
3500	3500	10000	10000	10000	0	0	20
3500	3500	10000	10000	10000	10000	20	0
3500	3500	10000	10000	10000	10000	10000	0
3500	3500	10000	10000	10000	10000	10000	10000
3500	3500	10000	10000	10000	10000	10000	10000
3500	3500	10000	10000	10000	10000	10000	10000
9000	9000	10000	10000	10000	10000	10000	10000
9000	9000	0	10000	10000	10000	10000	10000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000

9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000

9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
72	**73**	**74**	**75**	**76**	**77**	**78**	**79**
9000	**20**	**0**	**20**	**0**	**20**	**0**	**20**

79	80	81	82	83	84	85	86
30	30	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	3500	3500
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	9000	9000	3500	3500	3500	3500
30	30	30	30	30	30	30	30
3500	3500	9000	9000	3500	3500	3500	3500
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500

9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
10000	10000	10000	10000	10000	0	9000	9000
10000	10000	10000	10000	10000	10000	9000	9000
10000	10000	10000	10000	10000	10000	3500	3500
10000	10000	10000	10000	10000	10000	3500	3500
10000	10000	10000	10000	10000	10000	3500	3500
0	10000	10000	10000	10000	10000	3500	3500
0	20	10000	10000	10000	10000	3500	3500
20	0	0	10000	10000	10000	3500	3500
10000	0	0	20	10000	10000	3500	3500
10000	10000	20	0	0	10000	3500	3500
10000	10000	10000	0	0	20	9000	9000
10000	10000	10000	10000	20	0	9000	9000
3500	3500	3500	3500	9000	9000	0	20
3500	3500	3500	3500	9000	9000	20	0
3500	3500	9000	9000	3500	3500	10000	0
3500	3500	9000	9000	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000

3500	3500	9000	9000	3500	3500	10000	10000
3500	3500	9000	9000	3500	3500	10000	10000
3500	3500	3500	3500	9000	9000	10000	10000
3500	3500	3500	3500	9000	9000	0	10000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	9000	9000	30	30	30	30
30	30	9000	9000	30	30	30	30
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30

30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	30	30
30	30	9000	9000	30	30	30	30
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
80	**81**	**82**	**83**	**84**	**85**	**86**	**87**
0	**20**	**0**	**20**	**9000**	**20**	**0**	**20**

87	88	89	90	91	92	93	94
30	30	30	30	30	30	30	30
3500	3500	9000	9000	9000	9000	3500	3500
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	3500	3500
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	3500	3500
30	30	30	30	30	30	30	30
3500	3500	9000	9000	9000	9000	3500	3500
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500

9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
10000	10000	10000	10000	10000	10000	10000	10000
0	10000	10000	10000	10000	10000	10000	10000
0	20	10000	10000	10000	10000	10000	10000
20	0	0	10000	10000	10000	10000	10000
10000	0	0	20	10000	10000	10000	10000
10000	10000	20	0	0	10000	10000	10000
10000	10000	10000	0	0	20	10000	10000
10000	10000	10000	10000	20	0	0	10000

10000	10000	10000	10000	10000	0	0	20
10000	10000	10000	10000	10000	10000	20	0
10000	10000	10000	10000	10000	10000	10000	0
10000	10000	10000	10000	10000	10000	10000	10000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000

9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
88	**89**	**90**	**91**	**92**	**93**	**94**	**95**
0	**20**	**0**	**20**	**0**	**20**	**0**	**20**

95	96	97	98	99	100	101	102
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
9000	9000	3500	3500	9000	9000	3500	3500
30	30	30	30	30	30	30	30
3500	3500	9000	9000	3500	3500	3500	3500
30	30	30	30	30	30	30	30
3500	3500	9000	9000	3500	3500	3500	3500
30	30	30	30	30	30	30	30
9000	9000	3500	3500	9000	9000	3500	3500
30	30	30	30	30	30	30	30
3500	3500	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	9000	9000	30	30	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000

3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
10000	0	9000	9000	3500	3500	3500	3500
10000	10000	9000	9000	3500	3500	3500	3500
10000	10000	3500	3500	9000	9000	3500	3500
10000	10000	3500	3500	9000	9000	3500	3500
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000

10000	10000	3500	3500	9000	9000	3500	3500
0	10000	3500	3500	9000	9000	3500	3500
0	20	9000	9000	3500	3500	3500	3500
20	0	9000	9000	3500	3500	3500	3500
9000	9000	0	20	10000	10000	10000	10000
9000	9000	20	0	0	10000	10000	10000
3500	3500	10000	0	0	20	10000	10000
3500	3500	10000	10000	20	0	0	10000
3500	3500	10000	10000	10000	0	0	20
3500	3500	10000	10000	10000	10000	20	0
3500	3500	10000	10000	10000	10000	10000	0
3500	3500	10000	10000	10000	10000	10000	10000
3500	3500	10000	10000	10000	10000	10000	10000
3500	3500	10000	10000	10000	10000	10000	10000
9000	9000	10000	10000	10000	10000	10000	10000
9000	9000	0	10000	10000	10000	10000	10000
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000

9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000

96	**97**	**98**	**99**	**100**	**101**	**102**	**103**
9000	**20**	**0**	**20**	**0**	**20**	**0**	**20**

103	104	105	106	107	108	109	110
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	9000	9000	3500	3500	9000	9000
30	30	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	9000	9000
30	30	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	9000	9000
30	30	30	30	30	30	30	30
3500	3500	9000	9000	3500	3500	9000	9000
30	30	30	30	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000

9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000

3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
10000	10000	10000	10000	10000	0	30	30
10000	10000	10000	10000	10000	10000	30	30
10000	10000	10000	10000	10000	10000	30	30
10000	10000	10000	10000	10000	10000	30	30
10000	10000	10000	10000	10000	10000	9000	9000
0	10000	10000	10000	10000	10000	9000	9000
0	20	10000	10000	10000	10000	9000	9000
20	0	0	10000	10000	10000	9000	9000
10000	0	0	20	10000	10000	30	30
10000	10000	20	0	0	10000	30	30
10000	10000	10000	0	0	20	30	30
10000	10000	10000	10000	20	0	30	30
9000	9000	30	30	30	30	0	20
9000	9000	30	30	30	30	20	0
9000	9000	9000	9000	9000	9000	10000	0
9000	9000	9000	9000	9000	9000	10000	10000
9000	9000	30	30	30	30	10000	10000
9000	9000	30	30	30	30	10000	10000
9000	9000	30	30	30	30	10000	10000
9000	9000	30	30	30	30	10000	10000
9000	9000	9000	9000	9000	9000	10000	10000
9000	9000	9000	9000	9000	9000	10000	10000
9000	9000	30	30	30	30	10000	10000
9000	9000	30	30	30	30	0	10000
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500

3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	30	30	9000	9000	9000	9000
30	30	30	30	9000	9000	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
104	**105**	**106**	**107**	**108**	**109**	**110**	**111**
0	**20**	**0**	**20**	**30**	**20**	**0**	**20**

111	112	113	114	115	116	117	118
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000

9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000

30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
10000	10000	10000	10000	10000	10000	10000	10000
0	10000	10000	10000	10000	10000	10000	10000
0	20	10000	10000	10000	10000	10000	10000
20	0	0	10000	10000	10000	10000	10000
10000	0	0	20	10000	10000	10000	10000
10000	10000	20	0	0	10000	10000	10000
10000	10000	10000	0	0	20	10000	10000
10000	10000	10000	10000	20	0	0	10000
10000	10000	10000	10000	10000	0	0	20
10000	10000	10000	10000	10000	10000	20	0
10000	10000	10000	10000	10000	10000	10000	0
10000	10000	10000	10000	10000	10000	10000	10000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000

3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
112	**113**	**114**	**115**	**116**	**117**	**118**	**119**
0	**20**	**0**	**20**	**0**	**20**	**0**	**20**

119	120	121	122	123	124	125	126
30	30	30	30	30	30	30	30
9000	9000	9000	9000	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
9000	9000	9000	9000	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
9000	9000	9000	9000	30	30	30	30
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500

30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000

30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
10000	0	9000	9000	3500	3500	3500	3500
10000	10000	9000	9000	3500	3500	3500	3500
10000	10000	3500	3500	9000	9000	3500	3500
10000	10000	3500	3500	9000	9000	3500	3500
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	9000	9000	3500	3500
0	10000	3500	3500	9000	9000	3500	3500
0	20	9000	9000	3500	3500	3500	3500
20	0	9000	9000	3500	3500	3500	3500
9000	9000	0	20	10000	10000	10000	10000
9000	9000	20	0	0	10000	10000	10000
3500	3500	10000	0	0	20	10000	10000
3500	3500	10000	10000	20	0	0	10000
3500	3500	10000	10000	10000	0	0	20
3500	3500	10000	10000	10000	10000	20	0
3500	3500	10000	10000	10000	10000	10000	0
3500	3500	10000	10000	10000	10000	10000	10000
3500	3500	10000	10000	10000	10000	10000	10000
3500	3500	10000	10000	10000	10000	10000	10000
9000	9000	10000	10000	10000	10000	10000	10000
9000	9000	0	10000	10000	10000	10000	10000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000

3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
120	**121**	**122**	**123**	**124**	**125**	**126**	**127**
9000	**20**	**0**	**20**	**0**	**20**	**0**	**20**

127	128	129	130	131	132	133	134
30	30	30	30	30	30	30	30
30	30	30	30	9000	9000	9000	9000
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	9000	9000	9000	9000
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	9000	9000	9000	9000
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
30	30	9000	9000	30	30	30	30
30	30	9000	9000	30	30	30	30
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	30	30
30	30	9000	9000	30	30	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30

9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30

3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
10000	10000	10000	10000	10000	0	9000	9000
10000	10000	10000	10000	10000	10000	9000	9000
10000	10000	10000	10000	10000	10000	3500	3500
10000	10000	10000	10000	10000	10000	3500	3500
10000	10000	10000	10000	10000	10000	3500	3500
0	10000	10000	10000	10000	10000	3500	3500
0	20	10000	10000	10000	10000	3500	3500
20	0	0	10000	10000	10000	3500	3500
10000	0	0	20	10000	10000	3500	3500
10000	10000	20	0	0	10000	3500	3500
10000	10000	10000	0	0	20	9000	9000
10000	10000	10000	10000	20	0	9000	9000
3500	3500	3500	3500	9000	9000	0	20
3500	3500	3500	3500	9000	9000	20	0
3500	3500	9000	9000	3500	3500	10000	0
3500	3500	9000	9000	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000

3500	3500	9000	9000	3500	3500	10000	10000
3500	3500	9000	9000	3500	3500	10000	10000
3500	3500	3500	3500	9000	9000	10000	10000
3500	3500	3500	3500	9000	9000	0	10000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000

128	**129**	**130**	**131**	**132**	**133**	**134**	**135**
0	**20**	**0**	**20**	**9000**	**20**	**0**	**20**

135	136	137	138	139	140	141	142
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30

9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000

9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
10000	10000	10000	10000	10000	10000	10000	10000
0	10000	10000	10000	10000	10000	10000	10000
0	20	10000	10000	10000	10000	10000	10000
20	0	0	10000	10000	10000	10000	10000
10000	0	0	20	10000	10000	10000	10000
10000	10000	20	0	0	10000	10000	10000
10000	10000	10000	0	0	20	10000	10000
10000	10000	10000	10000	20	0	0	10000

10000	10000	10000	10000	10000	0	0	20
10000	10000	10000	10000	10000	10000	20	0
10000	10000	10000	10000	10000	10000	10000	0
10000	10000	10000	10000	10000	10000	10000	10000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500

136	**137**	**138**	**139**	**140**	**141**	**142**	**143**
0	**20**	**0**	**20**	**0**	**20**	**0**	**20**

143	144	145	146	147	148	149	150
30	30	30	30	30	30	30	30
9000	9000	9000	9000	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
9000	9000	9000	9000	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	30	30	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500

9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000

9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
10000	0	9000	9000	9000	9000	9000	9000
10000	10000	9000	9000	9000	9000	9000	9000
10000	10000	9000	9000	30	30	30	30
10000	10000	9000	9000	30	30	30	30
10000	10000	9000	9000	30	30	30	30
10000	10000	9000	9000	30	30	30	30
10000	10000	9000	9000	30	30	30	30
10000	10000	9000	9000	30	30	30	30

10000	10000	9000	9000	30	30	30	30
0	10000	9000	9000	30	30	30	30
0	20	9000	9000	9000	9000	9000	9000
20	0	9000	9000	9000	9000	9000	9000
9000	9000	0	20	10000	10000	10000	10000
9000	9000	20	0	0	10000	10000	10000
9000	9000	10000	0	0	20	10000	10000
9000	9000	10000	10000	20	0	0	10000
9000	9000	10000	10000	10000	0	0	20
9000	9000	10000	10000	10000	10000	20	0
9000	9000	10000	10000	10000	10000	10000	0
9000	9000	10000	10000	10000	10000	10000	10000
9000	9000	10000	10000	10000	10000	10000	10000
9000	9000	10000	10000	10000	10000	10000	10000
9000	9000	10000	10000	10000	10000	10000	10000
9000	9000	0	10000	10000	10000	10000	10000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500

144	**145**	**146**	**147**	**148**	**149**	**150**	**151**
9000	**20**	**0**	**20**	**0**	**20**	**0**	**20**

151	152	153	154	155	156	157	158
30	30	30	30	30	30	30	30
30	30	30	30	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	9000	9000	9000	9000
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	9000	9000	9000	9000
30	30	9000	9000	30	30	30	30
30	30	9000	9000	30	30	30	30
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	30	30
30	30	9000	9000	30	30	30	30
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30
3500	3500	9000	9000	3500	3500	30	30

3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	3500	3500
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	30	30
9000	9000	30	30	30	30	30	30
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30
9000	9000	3500	3500	3500	3500	30	30

3500	3500	3500	3500	9000	9000	30	30
3500	3500	3500	3500	9000	9000	30	30
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	3500	3500
30	30	9000	9000	30	30	9000	9000
30	30	9000	9000	30	30	9000	9000
9000	9000	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500

30	30	30	30	9000	9000	3500	3500
30	30	30	30	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
10000	10000	10000	10000	10000	0	9000	9000
10000	10000	10000	10000	10000	10000	9000	9000
10000	10000	10000	10000	10000	10000	3500	3500
10000	10000	10000	10000	10000	10000	3500	3500
10000	10000	10000	10000	10000	10000	3500	3500
0	10000	10000	10000	10000	10000	3500	3500
0	20	10000	10000	10000	10000	3500	3500
20	0	0	10000	10000	10000	3500	3500
10000	0	0	20	10000	10000	3500	3500
10000	10000	20	0	0	10000	3500	3500
10000	10000	10000	0	0	20	9000	9000
10000	10000	10000	10000	20	0	9000	9000
3500	3500	3500	3500	9000	9000	0	20
3500	3500	3500	3500	9000	9000	20	0
3500	3500	9000	9000	3500	3500	10000	0
3500	3500	9000	9000	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000
9000	9000	3500	3500	3500	3500	10000	10000
3500	3500	9000	9000	3500	3500	10000	10000
3500	3500	9000	9000	3500	3500	10000	10000
3500	3500	3500	3500	9000	9000	10000	10000
3500	3500	3500	3500	9000	9000	0	10000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000	9000	9000

152	**153**	**154**	**155**	**156**	**157**	**158**	**159**
0	**20**	**0**	**20**	**9000**	**20**	**0**	**20**

159	160	161	162	163	164	165	166
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30

30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	30	30
30	30	9000	9000	9000	9000	30	30
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000

9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500

9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
10000	10000	10000	10000	10000	10000	10000	10000
0	10000	10000	10000	10000	10000	10000	10000
0	20	10000	10000	10000	10000	10000	10000
20	0	0	10000	10000	10000	10000	10000
10000	0	0	20	10000	10000	10000	10000
10000	10000	20	0	0	10000	10000	10000
10000	10000	10000	0	0	20	10000	10000
10000	10000	10000	10000	20	0	0	10000
10000	10000	10000	10000	10000	0	0	20
10000	10000	10000	10000	10000	10000	20	0
10000	10000	10000	10000	10000	10000	10000	0
10000	10000	10000	10000	10000	10000	10000	10000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
3500	3500	9000	9000	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	3500	3500

160	**161**	**162**	**163**	**164**	**165**	**166**	**167**
0	**20**	**0**	**20**	**0**	**20**	**0**	**20**

167	168	169	170	171	172	173	174
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	30	30	30	30	30	30
9000	9000	9000	9000	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	30	30	30	30
30	30	30	30	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
30	30	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	30	30	30	30	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500

30	30	3500	3500	9000	9000	3500	3500
30	30	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
30	30	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
9000	9000	30	30	30	30	9000	9000
30	30	9000	9000	9000	9000	9000	9000
30	30	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
9000	9000	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	30	30	9000	9000	30	30
3500	3500	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	9000	9000	9000	9000
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500
30	30	9000	9000	3500	3500	3500	3500

30	30	3500	3500	3500	3500	9000	9000
30	30	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
3500	3500	9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500

3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	3500	3500	3500	3500
10000	0	9000	9000	3500	3500	3500	3500
10000	10000	9000	9000	3500	3500	3500	3500
10000	10000	3500	3500	9000	9000	3500	3500
10000	10000	3500	3500	9000	9000	3500	3500
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	3500	3500	9000	9000
10000	10000	3500	3500	9000	9000	3500	3500
0	10000	3500	3500	9000	9000	3500	3500
0	20	9000	9000	3500	3500	3500	3500
20	0	9000	9000	3500	3500	3500	3500
9000	9000	0	20	10000	10000	10000	10000
9000	9000	20	0	0	10000	10000	10000
3500	3500	10000	0	0	20	10000	10000
3500	3500	10000	10000	20	0	0	10000
3500	3500	10000	10000	10000	0	0	20
3500	3500	10000	10000	10000	10000	20	0
3500	3500	10000	10000	10000	10000	10000	0
3500	3500	10000	10000	10000	10000	10000	10000
3500	3500	10000	10000	10000	10000	10000	10000
3500	3500	10000	10000	10000	10000	10000	10000
9000	9000	10000	10000	10000	10000	10000	10000
9000	9000	0	10000	10000	10000	10000	10000
168	**169**	**170**	**171**	**172**	**173**	**174**	**175**
9000	**20**	**0**	**20**	**0**	**20**	**0**	**20**

175	176	177	178	179	180
30	30	30	30	30	30
9000	9000	9000	9000	9000	9000
30	30	30	30	30	30
30	30	30	30	9000	9000
30	30	30	30	30	30
30	30	30	30	9000	9000
30	30	30	30	30	30
30	30	30	30	9000	9000
30	30	30	30	30	30
30	30	30	30	9000	9000
30	30	30	30	30	30
9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000
30	30	9000	9000	30	30
30	30	9000	9000	30	30
30	30	9000	9000	30	30
30	30	9000	9000	30	30
30	30	9000	9000	30	30
30	30	9000	9000	30	30
30	30	9000	9000	30	30
30	30	9000	9000	30	30
9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30
9000	9000	30	30	30	30
9000	9000	30	30	30	30
9000	9000	30	30	30	30
9000	9000	30	30	30	30
9000	9000	30	30	30	30
9000	9000	30	30	30	30
9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000

3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000
9000	9000	30	30	30	30
9000	9000	30	30	30	30
9000	9000	30	30	30	30
9000	9000	30	30	30	30
9000	9000	30	30	30	30
9000	9000	30	30	30	30
9000	9000	30	30	30	30
9000	9000	30	30	30	30
9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000
30	30	9000	9000	30	30
30	30	9000	9000	30	30
30	30	9000	9000	30	30
30	30	9000	9000	30	30
30	30	9000	9000	30	30
30	30	9000	9000	30	30
30	30	9000	9000	30	30
30	30	9000	9000	30	30
9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000

9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000
30	30	30	30	9000	9000
30	30	30	30	9000	9000
30	30	30	30	9000	9000
30	30	30	30	9000	9000
30	30	30	30	9000	9000
30	30	30	30	9000	9000
30	30	30	30	9000	9000
30	30	30	30	9000	9000
9000	9000	9000	9000	9000	9000
9000	9000	9000	9000	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500

9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
9000	9000	3500	3500	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	9000	9000	3500	3500
3500	3500	3500	3500	9000	9000
3500	3500	3500	3500	9000	9000
10000	10000	10000	10000	10000	0
10000	10000	10000	10000	10000	10000
10000	10000	10000	10000	10000	10000
10000	10000	10000	10000	10000	10000
10000	10000	10000	10000	10000	10000
0	10000	10000	10000	10000	10000
0	20	10000	10000	10000	10000
20	0	0	10000	10000	10000
10000	0	0	20	10000	10000
10000	10000	20	0	0	10000
10000	10000	10000	0	0	20
10000	10000	10000	10000	20	0
176	**177**	**178**	**179**	**180**	**1**
0	**20**	**0**	**20**	**30**	

SOLUCIÓN DEL DANTZIG42 USANDO INDICE Y SOLVER

42

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	8	39	37	50	61	58	59	62	81	103	108	145
2	8	0	45	47	49	62	60	60	66	81	107	117	149
3	39	45	0	9	21	21	16	15	20	40	62	66	104
4	37	47	9	0	15	20	17	20	25	44	67	71	108
5	50	49	21	15	0	17	18	26	31	50	72	77	114
6	61	62	21	20	17	0	6	17	22	41	63	68	106
7	58	60	16	17	18	6	0	10	15	35	57	61	99
8	59	60	15	20	26	17	10	0	5	24	46	51	88
9	62	66	20	25	31	22	15	5	0	20	41	46	84
10	81	81	40	44	50	41	35	24	20	0	23	26	63
11	103	107	62	67	72	63	57	46	41	23	0	11	49
12	108	117	66	71	77	68	61	51	46	26	11	0	40
13	145	149	104	108	114	106	99	88	84	63	49	40	0
14	181	185	140	144	150	142	135	124	120	99	85	76	35
15	187	191	146	150	156	142	137	130	125	105	90	81	41
16	161	170	120	124	130	115	110	104	105	90	72	62	34
17	142	146	101	104	111	97	91	85	86	75	51	59	29
18	174	178	133	138	143	129	123	117	118	107	83	84	54
19	185	186	142	143	140	130	126	124	128	118	93	101	72
20	164	165	120	123	124	106	106	105	110	104	86	97	71
21	137	139	94	96	94	80	78	77	84	77	56	64	65
22	117	122	77	80	83	68	62	60	61	50	34	42	49
23	114	118	73	78	84	69	63	57	59	48	28	36	43
24	85	89	44	48	53	41	34	28	29	22	23	35	69
25	77	80	36	40	46	34	27	19	21	14	29	40	77
26	87	89	44	46	46	30	28	29	32	27	36	47	78
27	91	93	48	50	48	34	32	33	36	30	34	45	77
28	105	106	62	63	64	47	46	49	54	48	46	59	85
29	111	113	69	71	66	51	53	56	61	57	59	71	96
30	91	92	50	51	46	30	34	38	43	49	60	71	103
31	83	85	42	43	38	22	26	32	36	51	63	75	106
32	89	91	55	55	50	34	39	44	49	63	76	87	120
33	95	97	64	63	56	42	49	56	60	75	86	97	126
34	74	81	44	43	35	23	30	39	44	62	78	89	121
35	67	69	42	41	31	25	32	41	46	64	83	90	130
36	74	76	61	60	42	44	51	60	66	83	102	110	147
37	57	59	46	41	25	30	36	47	52	71	93	98	136
38	45	46	41	34	20	34	38	48	53	73	96	99	137
39	35	37	35	26	18	34	36	46	51	70	93	97	134
40	29	33	30	21	18	35	33	40	45	65	87	91	117
41	3	11	41	37	47	57	55	58	63	83	105	109	147
42	5	12	55	41	53	64	61	61	66	84	111	113	150

VECTOR													
1	2	3	4	5	6	7	8	9	10	11	12	13	14
DISTANCIAS													
8	45	9	15	17	6	10	5	20	23	11	40	35	10
SOLUCIÓN													
699													

14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
181	187	161	142	174	185	164	137	117	114	85	77	87	91	105
185	191	170	146	178	186	165	139	122	118	89	80	89	93	106
140	146	120	101	133	142	120	94	77	73	44	36	44	48	62
144	150	124	104	138	143	123	96	80	78	48	40	46	50	63
150	156	130	111	143	140	124	94	83	84	53	46	46	48	64
142	142	115	97	129	130	106	80	68	69	41	34	30	34	47
135	137	110	91	123	126	106	78	62	63	34	27	28	32	46
124	130	104	85	117	124	105	77	60	57	28	19	29	33	49
120	125	105	86	118	128	110	84	61	59	29	21	32	36	54
99	105	90	75	107	118	104	77	50	48	22	14	27	30	48
85	90	72	51	83	93	86	56	34	28	23	29	36	34	46
76	81	62	59	84	101	97	64	42	36	35	40	47	45	59
35	41	34	29	54	72	71	65	49	43	69	77	78	77	85
0	10	31	53	46	69	93	90	82	77	105	114	116	115	119
10	0	27	48	35	58	82	87	77	72	102	111	112	110	115
31	27	0	21	26	58	62	58	60	45	74	84	84	83	88
53	48	21	0	31	43	42	36	30	27	56	64	66	63	66
46	35	26	31	0	26	45	68	62	59	88	96	98	97	98
69	58	58	43	26	0	22	50	70	69	99	107	95	91	79
93	82	62	42	45	22	0	30	49	55	81	87	75	72	59
90	87	58	36	68	50	30	0	21	27	54	60	47	44	31
82	77	60	30	62	70	49	21	0	5	32	40	36	32	36
77	72	45	27	59	69	55	27	5	0	29	37	39	36	42
105	102	74	56	88	99	81	54	32	29	0	8	12	9	28
114	111	84	64	96	107	87	60	40	37	8	0	11	15	33
116	112	84	66	98	95	75	47	36	39	12	11	0	3	21
115	110	83	63	97	91	72	44	32	36	9	15	3	0	20
119	115	88	66	98	79	59	31	36	42	28	33	21	20	0
130	126	98	75	98	85	62	38	47	53	39	42	29	30	12
141	136	109	90	115	99	81	53	61	62	36	34	24	28	20
142	140	112	93	126	108	88	60	64	66	39	36	27	31	28
155	150	123	100	123	109	86	62	71	78	52	49	39	44	35
160	155	128	104	128	113	90	67	76	82	62	59	49	53	40
159	155	127	108	136	124	101	75	79	81	54	50	42	46	43
164	160	133	114	146	134	111	85	84	86	59	52	47	51	53
185	179	155	133	159	146	122	98	105	107	79	71	66	70	70
172	172	148	126	158	147	124	121	97	99	71	65	59	63	67
176	178	151	131	163	159	135	108	102	103	73	67	64	69	75
171	176	151	129	161	163	139	118	102	101	71	65	65	70	84
166	171	144	125	157	156	139	113	95	97	67	60	62	67	79
186	188	164	144	176	182	161	134	119	116	86	78	84	88	101
186	192	166	147	180	188	167	140	124	119	90	87	90	94	107

15	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**
27	**21**	**31**	**26**	**22**	**30**	**21**	**5**	**29**	**8**	**11**	**3**	**20**	**12**	**20**

29	30	31	32	33	34	35	36	37	38	39	40	41	42
111	91	83	89	95	74	67	74	57	45	35	29	3	5
113	92	85	91	97	81	69	76	59	46	37	33	11	12
69	50	42	55	64	44	42	61	46	41	35	30	41	55
71	51	43	55	63	43	41	60	41	34	26	21	37	41
66	46	38	50	56	35	31	42	25	20	18	18	47	53
51	30	22	34	42	23	25	44	30	34	34	35	57	64
53	34	26	39	49	30	32	51	36	38	36	33	55	61
56	38	32	44	56	39	41	60	47	48	46	40	58	61
61	43	36	49	60	44	46	66	52	53	51	45	63	66
57	49	51	63	75	62	64	83	71	73	70	65	83	84
59	60	63	76	86	78	83	102	93	96	93	87	105	111
71	71	75	87	97	89	90	110	98	99	97	91	109	113
96	103	106	120	126	121	130	147	136	137	134	117	147	150
130	141	142	155	160	159	164	185	172	176	171	166	186	186
126	136	140	150	155	155	160	179	172	178	176	171	188	192
98	109	112	123	128	127	133	155	148	151	151	144	164	166
75	90	93	100	104	108	114	133	126	131	129	125	144	147
98	115	126	123	128	136	146	159	158	163	161	157	176	180
85	99	108	109	113	124	134	146	147	159	163	156	182	188
62	81	88	86	90	101	111	122	124	135	139	139	161	167
38	53	60	62	67	75	85	98	121	108	118	113	134	140
47	61	64	71	76	79	84	105	97	102	102	95	119	124
53	62	66	78	82	81	86	107	99	103	101	97	116	119
39	36	39	52	62	54	59	79	71	73	71	67	86	90
42	34	36	49	59	50	52	71	65	67	65	60	78	87
29	24	27	39	49	42	47	66	59	64	65	62	84	90
30	28	31	44	53	46	51	70	63	69	70	67	88	94
12	20	28	35	40	43	53	70	67	75	84	79	101	107
0	20	28	24	29	39	49	60	62	72	78	82	108	114
20	0	8	15	25	23	32	48	46	54	58	62	88	77
28	8	0	12	23	14	24	40	38	46	50	53	80	86
24	15	12	0	11	14	24	36	37	49	56	59	86	92
29	25	23	11	0	21	30	33	43	54	62	66	92	98
39	23	14	14	21	0	9	25	23	34	41	45	71	80
49	32	24	24	30	9	0	18	13	24	32	38	64	74
60	48	40	36	33	25	18	0	17	29	38	45	71	77
62	46	38	37	43	23	13	17	0	12	21	27	54	60
72	54	46	49	54	34	24	29	12	0	9	15	41	48
78	58	50	56	62	41	32	38	21	9	0	6	32	38
82	62	53	59	66	45	38	45	27	15	6	0	25	32
108	88	80	86	92	71	64	71	54	41	32	25	0	6
114	77	86	92	98	80	74	77	60	48	38	32	6	0

30	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**1**
8	**12**	**11**	**21**	**9**	**18**	**17**	**12**	**9**	**6**	**25**	**6**	**5**	

SOLUCIÓN DEL FRI26 USANDO INDICE Y SOLVER

26

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	83	93	129	133	139	151	169	135	114	110	98	99
2	83	0	40	53	62	64	91	116	93	84	95	98	89
3	93	40	0	42	42	49	59	81	54	44	58	64	54
4	129	53	42	0	11	11	46	72	65	70	88	100	89
5	133	62	42	11	0	9	35	61	55	62	82	95	84
6	139	64	49	11	9	0	39	65	63	71	90	103	92
7	151	91	59	46	35	39	0	26	34	52	71	88	77
8	169	116	81	72	61	65	26	0	37	59	75	92	83
9	135	93	54	65	55	63	34	37	0	22	39	56	47
10	114	84	44	70	62	71	52	59	22	0	20	36	26
11	110	95	58	88	82	90	71	75	39	20	0	18	11
12	98	98	64	100	95	103	88	92	56	36	18	0	11
13	99	89	54	89	84	92	77	83	47	26	11	11	0
14	95	68	31	66	62	71	63	76	40	20	27	34	23
15	81	67	36	76	74	82	78	91	55	34	32	31	24
16	152	127	86	102	93	100	66	54	37	43	42	56	53
17	159	156	117	142	133	141	110	98	78	74	61	63	68
18	181	175	135	156	146	153	119	103	91	91	80	85	89
19	172	152	112	127	117	124	88	70	62	68	64	75	74
20	185	165	125	139	128	135	98	78	74	82	77	87	87
21	147	160	124	155	148	156	130	122	96	86	68	62	71
22	157	180	147	180	173	181	156	148	122	111	92	83	93
23	185	223	193	228	222	230	206	198	172	160	140	129	140
24	220	268	241	278	272	280	257	250	223	210	190	178	189
25	127	179	157	197	194	202	188	188	155	136	116	100	111
26	181	197	161	190	182	190	160	148	128	121	103	99	107

VECTOR

12 25 24 23 26 22 21 17 18 20 19 16 9 8

DISTANCIAS

100 93 51 58 27 26 30 22 29 13 26 37 37 26

SOLUCIÓN

953

14	15	16	17	18	19	20	21	22	23	24	25	26
95	81	152	159	181	172	185	147	157	185	220	127	181
68	67	127	156	175	152	165	160	180	223	268	179	197
31	36	86	117	135	112	125	124	147	193	241	157	161
66	76	102	142	156	127	139	155	180	228	278	197	190
62	74	93	133	146	117	128	148	173	222	272	194	182
71	82	100	141	153	124	135	156	181	230	280	202	190
63	78	66	110	119	88	98	130	156	206	257	188	160
76	91	54	98	103	70	78	122	148	198	250	188	148
40	55	37	78	91	62	74	96	122	172	223	155	128
20	34	43	74	91	68	82	86	111	160	210	136	121
27	32	42	61	80	64	77	68	92	140	190	116	103
34	31	56	63	85	75	87	62	83	129	178	100	99
23	24	53	68	89	74	87	71	93	140	189	111	107
0	15	62	87	106	87	100	93	116	163	212	132	130
15	0	73	92	112	96	109	93	113	158	205	122	130
62	73	0	44	54	26	39	68	94	144	196	139	95
87	92	44	0	22	34	38	30	53	102	154	109	51
106	112	54	22	0	33	29	46	64	107	157	125	51
87	96	26	34	33	0	13	63	87	135	186	141	81
100	109	39	38	29	13	0	68	90	136	186	148	79
93	93	68	30	46	63	68	0	26	77	128	80	37
116	113	94	53	64	87	90	26	0	50	102	65	27
163	158	144	102	107	135	136	77	50	0	51	64	58
212	205	196	154	157	186	186	128	102	51	0	93	107
132	122	139	109	125	141	148	80	65	64	93	0	90
130	130	95	51	51	81	79	37	27	58	107	90	0
7	**5**	**6**	**4**	**3**	**2**	**1**	**15**	**14**	**10**	**11**	**13**	12
35	**9**	**11**	**42**	**40**	**83**	**81**	**15**	**20**	**20**	**11**	**11**	

SOLUCIÓN DEL GR17 USANDO INDICE Y SOLVER

17

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	633	257	91	412	150	80	134	259	505	353	324	70
2	633	0	390	661	227	488	572	530	555	289	282	638	567
3	257	390	0	228	169	112	196	154	372	262	110	437	191
4	91	661	228	0	383	120	77	105	175	476	324	240	27
5	412	227	169	383	0	267	351	309	338	196	61	421	346
6	150	488	112	120	267	0	63	34	264	360	208	329	83
7	80	572	196	77	351	63	0	29	232	444	292	297	47
8	134	530	154	105	309	34	29	0	249	402	250	314	68
9	259	555	372	175	338	264	232	249	0	495	352	95	189
10	505	289	262	476	196	360	444	402	495	0	154	578	439
11	353	282	110	324	61	208	292	250	352	154	0	435	287
12	324	638	437	240	421	329	297	314	95	578	435	0	254
13	70	567	191	27	346	83	47	68	189	439	287	254	0
14	211	466	74	182	243	105	150	108	326	336	184	391	145
15	268	420	53	239	199	123	207	165	383	240	140	448	202
16	246	745	472	237	528	364	332	349	202	685	542	157	289
17	121	518	142	84	297	35	29	36	236	390	238	301	55

VECTOR

2	**10**	**15**	**14**	**17**	**13**	**4**	**9**	**12**	**16**	**1**	**7**	**8**	**6**

DISTANCIAS

289	**240**	**57**	**96**	**55**	**27**	**175**	**95**	**157**	**246**	**80**	**29**	**34**	**112**

SOLUCIÓN

2090

14	15	16	17
211	268	246	121
466	420	745	518
74	53	472	142
182	239	237	84
243	199	528	297
105	123	364	35
150	207	332	29
108	165	349	36
326	383	202	236
336	240	685	390
184	140	542	238
391	448	157	301
145	202	289	55
0	57	426	96
57	0	483	153
426	483	0	336
96	153	336	0
3	**11**	**5**	2
110	**61**	**227**	

SOLUCIÓN DEL GR21 USANDO INDICE Y SOLVER

21

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	510	635	91	385	155	110	130	490	370	155	68	610
2	510	0	355	415	585	475	480	500	605	320	380	440	360
3	635	355	0	605	390	495	570	540	295	700	640	575	705
4	91	415	605	0	350	120	78	97	460	280	63	27	520
5	385	585	390	350	0	240	320	285	120	590	430	320	835
6	155	475	495	120	240	0	96	36	350	365	200	91	605
7	110	480	570	78	320	96	0	29	425	350	160	48	590
8	130	500	540	97	285	36	29	0	390	370	175	67	610
9	490	605	295	460	120	350	425	390	0	625	535	430	865
10	370	320	700	280	590	365	350	370	625	0	240	300	250
11	155	380	640	63	430	200	160	175	535	240	0	90	480
12	68	440	575	27	320	91	48	67	430	300	90	0	545
13	610	360	705	520	835	605	590	610	865	250	480	545	0
14	655	235	585	555	750	615	625	645	775	285	515	585	190
15	480	81	435	380	575	440	455	465	600	245	345	415	295
16	265	480	420	235	125	125	200	165	230	475	310	205	715
17	255	440	755	235	650	370	320	350	680	150	175	265	400
18	450	270	625	345	660	430	420	440	690	77	310	380	180
19	170	445	750	160	495	265	220	240	600	235	125	170	485
20	240	290	590	140	480	255	205	220	515	150	100	170	390
21	380	140	495	280	480	340	350	370	505	185	240	310	345

VECTOR

4	**12**	**1**	**7**	**8**	**6**	**16**	**5**	**9**	**3**	**2**	**15**	**14**	**13**

DISTANCIAS

27	**68**	**110**	**29**	**36**	**125**	**125**	**120**	**295**	**355**	**81**	**170**	**190**	**180**

SOLUCIÓN

2803

14	15	16	17	18	19	20	21
655	480	265	255	450	170	240	380
235	81	480	440	270	445	290	140
585	435	420	755	625	750	590	495
555	380	235	235	345	160	140	280
750	575	125	650	660	495	480	480
615	440	125	370	430	265	255	340
625	455	200	320	420	220	205	350
645	465	165	350	440	240	220	370
775	600	230	680	690	600	515	505
285	245	475	150	77	235	150	185
515	345	310	175	310	125	100	240
585	415	205	265	380	170	170	310
190	295	715	400	180	485	390	345
0	170	650	435	215	525	425	280
170	0	475	385	190	405	255	105
650	475	0	485	545	375	395	380
435	385	485	0	225	87	205	280
215	190	545	225	0	315	220	165
525	405	375	87	315	0	155	305
425	255	395	205	220	155	0	150
280	105	380	280	165	305	150	0
18	**10**	**21**	**20**	**19**	**17**	**11**	4
77	**185**	**150**	**155**	**87**	**175**	**63**	

SOLUCIÓN DEL GR24 USANDO INDICE Y SOLVER

24

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	257	187	91	150	80	130	134	243	185	214	70	272
2	257	0	196	228	112	196	167	154	209	86	223	191	180
3	187	196	0	158	96	88	59	63	286	124	49	121	315
4	91	228	158	0	120	77	101	105	159	156	185	27	188
5	150	112	96	120	0	63	56	34	190	40	123	83	193
6	80	196	88	77	63	0	25	29	216	124	115	47	245
7	130	167	59	101	56	25	0	22	229	95	86	64	258
8	134	154	63	105	34	29	22	0	225	82	90	68	228
9	243	209	286	159	190	216	229	225	0	207	313	173	29
10	185	86	124	156	40	124	95	82	207	0	151	119	159
11	214	223	49	185	123	115	86	90	313	151	0	148	342
12	70	191	121	27	83	47	64	68	173	119	148	0	209
13	272	180	315	188	193	245	258	228	29	159	342	209	0
14	219	83	172	149	79	139	134	112	126	62	199	153	97
15	293	50	232	264	148	232	203	190	248	122	259	227	219
16	54	219	92	82	119	31	43	58	238	147	84	53	267
17	211	74	81	182	105	150	121	108	310	37	160	145	196
18	290	139	98	261	144	176	164	136	389	116	147	224	275
19	268	53	138	239	123	207	178	165	367	86	187	202	227
20	261	43	200	232	98	200	171	131	166	90	227	195	137
21	175	128	76	146	32	76	47	30	222	56	103	109	225
22	250	99	89	221	105	189	160	147	349	76	138	184	235
23	192	228	235	108	119	165	178	154	71	136	262	110	74
24	121	142	99	84	35	29	42	36	220	70	126	55	249

VECTOR

4 1 16 11 3 7 8 21 5 10 17 18 22 19

DISTANCIAS

91 54 84 49 59 22 30 32 40 37 79 40 46 53

SOLUCIÓN

1388

14	15	16	17	18	19	20	21	22	23	24
219	293	54	211	290	268	261	175	250	192	121
83	50	219	74	139	53	43	128	99	228	142
172	232	92	81	98	138	200	76	89	235	99
149	264	82	182	261	239	232	146	221	108	84
79	148	119	105	144	123	98	32	105	119	35
139	232	31	150	176	207	200	76	189	165	29
134	203	43	121	164	178	171	47	160	178	42
112	190	58	108	136	165	131	30	147	154	36
126	248	238	310	389	367	166	222	349	71	220
62	122	147	37	116	86	90	56	76	136	70
199	259	84	160	147	187	227	103	138	262	126
153	227	53	145	224	202	195	109	184	110	55
97	219	267	196	275	227	137	225	235	74	249
0	134	170	99	178	130	69	104	138	96	104
134	0	255	125	154	68	82	164	114	264	178
170	255	0	173	190	230	223	99	212	187	60
99	125	173	0	79	57	90	57	39	182	96
178	154	190	79	0	86	176	112	40	261	175
130	68	230	57	86	0	90	114	46	239	153
69	82	223	90	176	90	0	134	136	165	146
104	164	99	57	112	114	134	0	96	151	47
138	114	212	39	40	46	136	96	0	221	135
96	264	187	182	261	239	165	151	221	0	169
104	178	60	96	175	153	146	47	135	169	0
2	**15**	**20**	**13**	**9**	**23**	**14**	**24**	**6**	**12**	4
50	**82**	**137**	**29**	**71**	**96**	**104**	**29**	**47**	**27**	

SOLUCIÓN DEL GR48 USANDO INDICE Y SOLVER

48

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	593	409	566	633	257	91	412	378	593	150	659	80
2	593	0	258	331	586	602	509	627	755	416	598	488	566
3	409	258	0	171	723	522	325	506	634	564	414	630	382
4	566	331	171	0	874	679	482	663	791	721	571	787	539
5	633	586	723	874	0	390	598	227	397	271	488	205	572
6	257	602	522	679	390	0	228	169	175	445	112	511	196
7	91	509	325	482	598	228	0	383	349	509	120	575	77
8	412	627	506	663	227	169	383	0	167	293	267	304	351
9	378	755	634	791	397	175	349	167	0	429	233	470	317
10	593	416	564	721	271	445	509	293	429	0	541	76	563
11	150	598	414	571	488	112	120	267	233	541	0	607	63
12	659	488	630	787	205	511	575	304	470	76	607	0	629
13	80	566	382	539	572	196	77	351	317	563	63	629	0
14	434	893	699	856	524	231	405	303	138	595	289	606	373
15	455	417	433	590	313	304	371	228	394	158	399	224	425
16	134	583	399	566	530	154	105	309	275	575	34	638	29
17	649	945	824	981	446	423	620	357	280	649	504	648	588
18	259	364	180	337	555	272	175	338	466	403	264	469	232
19	505	354	110	70	819	618	421	602	730	660	509	728	478
20	710	117	375	354	679	693	626	720	848	533	715	610	683
21	488	784	663	820	289	262	459	196	119	488	343	502	427
22	353	641	520	677	282	110	324	61	125	353	208	364	292
23	324	275	91	248	638	437	240	421	549	486	329	552	297
24	605	287	431	588	313	445	520	470	598	143	610	215	577
25	372	229	39	196	686	485	288	469	597	511	397	578	345
26	330	484	361	518	378	119	260	150	278	323	174	389	276
27	581	877	756	913	370	355	552	289	212	581	436	571	520
28	154	460	276	433	612	298	63	453	419	460	190	526	158
29	70	523	339	496	569	191	27	346	312	516	83	589	47
30	606	183	216	147	715	719	522	703	831	549	611	615	579
31	585	427	563	720	179	437	501	196	362	80	532	108	558
32	544	840	719	876	311	318	515	252	175	508	399	494	483
33	496	525	595	751	147	253	468	85	251	208	351	236	435
34	317	289	105	262	631	430	233	414	542	479	332	545	290
35	648	68	316	362	584	598	564	625	753	418	653	484	621
36	211	660	476	633	466	74	182	243	171	489	66	555	150
37	475	137	295	452	437	428	391	452	580	271	480	337	448
38	654	151	319	266	755	767	570	751	879	561	659	627	627
39	710	239	487	546	616	660	626	687	815	443	715	509	683
40	585	135	385	458	499	535	501	562	690	333	590	399	558
41	246	373	183	340	745	472	237	528	656	593	364	659	332
42	788	208	456	488	724	738	704	765	893	558	793	624	761
43	446	162	111	268	624	559	362	543	671	458	451	524	419
44	166	437	247	404	749	435	150	590	556	597	402	663	295

45	523	81	188	255	596	636	439	620	648	430	528	496	496
46	235	371	187	344	581	348	151	364	469	429	240	495	208
47	369	205	289	446	537	328	286	355	483	371	375	437	343
48	121	570	386	543	518	142	84	297	263	570	35	636	29

VECTOR													
44	**41**	**23**	**3**	**19**	**4**	**30**	**20**	**38**	**25**	**43**	**40**	**39**	**42**
DISTANCIAS													
80	**157**	**91**	**110**	**70**	**147**	**209**	**103**	**290**	**82**	**289**	**121**	**138**	**138**
SOLUCIÓN													
6557													

14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
434	455	134	649	259	505	710	488	353	324	605	372	330	581	154
893	417	583	945	364	354	117	784	641	275	287	229	484	877	460
699	433	399	824	180	110	375	663	520	91	431	39	361	756	276
856	590	566	981	337	70	354	820	677	248	588	196	518	913	433
524	313	530	446	555	819	679	289	282	638	313	686	378	370	612
231	304	154	423	272	618	693	262	110	437	445	485	119	355	298
405	371	105	620	175	421	626	459	324	240	520	288	260	552	63
303	228	309	357	338	602	720	196	61	421	470	469	150	289	453
138	394	275	280	466	730	848	119	125	549	598	597	278	212	419
595	158	575	649	403	660	533	488	353	486	143	511	323	581	460
289	399	34	504	264	509	715	343	208	329	610	397	174	436	190
606	224	638	648	469	728	610	502	364	552	215	578	389	571	526
373	425	29	588	232	478	683	427	292	297	577	345	276	520	158
0	530	298	416	549	795	986	255	261	614	734	662	414	348	475
530	0	434	584	265	529	534	423	288	348	144	396	185	516	322
298	434	0	546	249	494	700	385	250	314	595	361	207	478	175
416	584	546	0	656	920	1038	161	315	739	788	787	468	84	690
549	265	249	656	0	276	481	495	352	95	352	143	193	588	126
795	529	494	920	276	0	345	759	616	187	527	135	475	852	372
986	534	700	1038	481	345	0	877	734	392	404	346	577	970	577
255	423	385	161	495	759	877	0	154	578	627	626	307	93	529
261	288	250	315	352	616	734	154	0	435	484	483	164	247	396
614	348	314	739	95	187	392	578	435	0	385	54	276	671	191
734	144	595	788	352	527	404	627	484	385	0	377	326	720	471
662	396	361	787	143	135	346	626	483	54	377	0	324	719	239
414	185	207	468	193	475	577	307	164	276	326	324	0	400	250
348	516	478	84	588	852	970	93	247	671	720	719	400	0	622
475	322	175	690	126	372	577	529	396	191	471	239	250	622	0
368	385	68	583	189	435	640	422	287	254	534	302	249	515	115
896	546	596	1021	377	139	209	860	717	288	416	242	558	953	473
498	163	567	552	395	659	544	391	256	478	154	526	318	484	452
311	479	441	154	551	815	933	65	210	634	683	682	363	77	585
387	162	393	441	427	691	646	280	145	509	249	558	239	373	538
607	341	307	732	88	201	406	571	428	21	407	68	269	664	184
891	415	638	943	395	412	95	782	639	333	285	287	482	875	515
227	351	108	432	326	572	777	271	184	391	492	439	166	364	252
718	268	465	770	222	391	254	609	466	255	138	241	309	702	342
944	558	644	1069	425	262	103	908	765	336	428	290	606	1001	521
953	440	700	1005	457	583	279	844	701	490	310	458	544	937	577
828	330	575	880	332	481	215	719	576	365	200	356	419	812	452
649	455	349	846	202	279	490	685	542	157	525	144	383	778	174
1031	555	778	1083	535	552	188	922	779	473	425	427	622	1015	655
736	455	436	861	217	207	279	700	557	128	325	82	398	793	313
612	459	387	827	189	343	554	666	531	221	589	208	372	759	137

813	427	513	938	294	284	193	777	634	205	297	159	475	870	390
525	291	225	682	32	283	488	521	378	103	384	150	219	614	94
554	269	360	673	116	385	322	512	369	149	238	230	209	605	237
319	432	36	534	236	482	687	373	238	301	581	349	222	466	162

35	**2**	**45**	**47**	**11**	**36**	**6**	**29**	**16**	**14**	**9**	**32**	**27**	**17**	**21**
68	**81**	**201**	**375**	**66**	**74**	**191**	**68**	**298**	**138**	**175**	**77**	**84**	**161**	**154**

29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
70	606	585	544	496	317	648	211	475	654	710	585	246	788	446
523	183	427	840	525	289	68	660	137	151	239	135	373	208	162
339	216	563	719	595	105	316	476	295	319	487	385	183	456	111
496	147	720	876	751	262	362	633	452	266	546	458	340	488	268
569	715	179	311	147	631	584	466	437	755	616	499	745	724	624
191	719	437	318	253	430	598	74	428	767	660	535	472	738	559
27	522	501	515	468	233	564	182	391	570	626	501	237	704	362
346	703	196	252	85	414	625	243	452	751	687	562	528	765	543
312	831	362	175	251	542	753	171	580	879	815	690	656	893	671
516	549	80	508	208	479	418	489	271	561	443	333	593	558	458
83	611	532	399	351	332	653	66	480	659	715	590	364	793	451
589	615	108	494	236	545	484	555	337	627	509	399	659	624	524
47	579	558	483	435	290	621	150	448	627	683	558	332	761	419
368	896	498	311	387	607	891	227	718	944	953	828	649	1031	736
385	546	163	479	162	341	415	351	268	558	440	330	455	555	455
68	596	567	441	393	307	638	108	465	644	700	575	349	778	436
583	1021	552	154	441	732	943	432	770	1069	1005	880	846	1083	861
189	377	395	551	427	88	395	326	222	425	457	332	202	535	217
435	139	659	815	691	201	412	572	391	262	583	481	279	552	207
640	209	544	933	646	406	95	777	254	103	279	215	490	188	279
422	860	391	65	280	571	782	271	609	908	844	719	685	922	700
287	717	256	210	145	428	639	184	466	765	701	576	542	779	557
254	288	478	634	509	21	333	391	255	336	490	365	157	473	128
534	416	154	683	249	407	285	492	138	428	310	200	525	425	325
302	242	526	682	558	68	287	439	241	290	458	356	144	427	82
249	558	318	363	239	269	482	166	309	606	544	419	383	622	398
515	953	484	77	373	664	875	364	702	1001	937	812	778	1015	793
115	473	452	585	538	184	515	252	342	521	577	452	174	655	313
0	536	515	479	430	247	578	145	405	584	640	515	289	718	376
536	0	556	916	654	302	209	673	287	122	393	318	386	343	175
515	556	0	399	128	471	425	438	278	568	450	340	585	565	465
479	916	399	0	336	627	838	327	665	964	900	775	741	978	756
430	654	128	336	0	503	523	327	376	666	548	438	618	663	563
247	302	471	627	503	0	347	384	277	350	512	387	132	487	142
578	209	425	838	523	347	0	715	167	169	179	120	431	138	220
145	673	438	327	327	384	715	0	542	721	777	652	426	855	513
405	287	278	665	376	277	167	542	0	299	229	104	395	307	187
584	122	568	964	666	350	169	721	299	0	353	289	434	284	223
640	393	450	900	548	512	179	777	229	353	0	121	630	138	391
515	318	340	775	438	387	120	652	104	289	121	0	505	235	289
289	386	585	741	618	132	431	426	395	434	630	505	0	571	226
718	343	565	978	663	487	138	855	307	284	138	235	571	0	360
376	175	465	756	563	142	220	513	187	223	391	289	226	360	0
177	450	589	722	675	196	495	389	459	498	694	569	80	635	290

453	119	437	833	535	219	139	590	168	131	310	208	303	279	92
165	384	421	577	454	92	429	302	254	432	489	364	165	569	224
300	352	378	568	445	172	281	436	108	332	343	218	290	421	164
55	583	562	429	381	294	625	96	452	631	687	562	336	765	423

22	**8**	**33**	**5**	**31**	**10**	**12**	**37**	**24**	**15**	**26**	**18**	**34**	**46**	**28**
61	**85**	**147**	**179**	**80**	**76**	**337**	**138**	**144**	**185**	**193**	**88**	**92**	**94**	**63**

44	45	46	47	48
166	523	235	369	121
437	81	371	205	570
247	188	187	289	386
404	255	344	446	543
749	596	581	537	518
435	636	348	328	142
150	439	151	286	84
590	620	364	355	297
556	648	469	483	263
597	430	429	371	570
402	528	240	375	35
663	496	495	437	636
295	496	208	343	29
612	813	525	554	319
459	427	291	269	432
387	513	225	360	36
827	938	682	673	534
189	294	32	116	236
343	284	283	385	482
554	193	488	322	687
666	777	521	512	373
531	634	378	369	238
221	205	103	149	301
589	297	384	238	581
208	159	150	230	349
372	475	219	209	222
759	870	614	605	466
137	390	94	237	162
177	453	165	300	55
450	119	384	352	583
589	437	421	378	562
722	833	577	568	429
675	535	454	445	381
196	219	92	172	294
495	139	429	281	625
389	590	302	436	96
459	168	254	108	452
498	131	432	332	631
694	310	489	343	687
569	208	364	218	562
80	303	165	290	336
635	279	569	421	765
290	92	224	164	423
0	367	154	354	299

367	0	301	201	500
154	301	0	149	212
354	201	149	0	347
299	500	212	347	0

7	**48**	**13**	**1**	44

84	**29**	**80**	**166**

SOLUCIÓN DEL GR120 USANDO INDICE Y SOLVER

120

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	534	434	294	593	409	332	232	464	566	552	802	633
2	534	0	107	241	190	351	320	354	124	508	80	316	432
3	434	107	0	148	137	240	232	261	88	397	127	336	479
4	294	241	148	0	374	190	139	113	171	347	259	509	552
5	593	190	137	374	0	258	494	372	202	331	234	222	586
6	409	351	240	190	258	0	310	188	328	171	365	470	723
7	332	320	232	139	494	310	0	208	188	467	249	588	417
8	232	354	261	113	372	188	208	0	284	345	372	584	621
9	464	124	88	171	202	328	188	284	0	485	61	392	411
10	566	508	397	347	331	171	467	345	485	0	522	502	874
11	552	80	127	259	234	365	249	372	61	522	0	386	354
12	802	316	336	509	222	470	588	584	392	502	386	0	738
13	633	432	479	552	586	723	417	621	411	874	354	738	0
14	257	641	541	407	706	522	184	391	372	679	433	915	390
15	187	577	477	337	636	452	375	321	507	609	595	845	572
16	91	450	357	210	509	325	248	141	380	482	468	718	661
17	412	624	531	384	690	506	210	408	398	663	459	892	227
18	400	752	659	512	818	634	338	536	526	791	587	1020	524
19	472	805	712	565	871	687	391	589	579	844	640	1073	413
20	389	665	572	425	731	547	251	449	439	704	500	933	274
21	610	76	183	317	192	442	396	430	202	515	141	233	492
22	340	730	630	490	789	605	394	474	582	762	643	998	444
23	510	152	134	217	248	370	175	330	46	527	72	438	380
24	153	447	354	207	470	280	246	113	377	437	465	715	659
25	511	844	751	604	910	726	430	628	618	883	679	1112	407
26	269	283	190	42	332	148	169	63	213	305	301	544	582
27	525	157	95	232	42	257	316	355	160	330	167	254	521
28	150	539	446	299	598	414	279	283	469	571	557	807	488
29	80	507	414	267	566	382	305	251	437	539	525	775	572
30	130	520	427	280	579	395	318	264	450	552	538	788	543
31	401	791	691	551	850	666	474	535	662	823	723	1059	524
32	134	524	431	284	583	399	314	268	454	556	542	792	530
33	666	942	849	702	1008	824	528	726	716	981	777	1210	446
34	259	281	188	40	364	180	142	72	211	337	299	549	555
35	505	447	336	286	354	110	406	284	424	70	461	566	819
36	453	358	247	234	265	59	354	232	335	182	372	477	767
37	627	334	251	408	168	239	528	406	300	166	348	331	700
38	339	275	187	94	313	281	45	184	143	438	204	543	458
39	710	283	254	491	117	375	611	489	319	354	351	202	679
40	243	353	260	113	419	235	89	133	283	392	368	621	502
41	376	520	427	280	586	402	106	300	294	559	355	788	340
42	449	594	501	354	660	476	180	378	368	633	429	862	256
43	505	781	688	541	847	663	367	565	555	820	616	1049	289
44	322	611	518	371	677	493	197	395	372	650	446	879	359
45	185	575	482	335	634	450	231	319	419	607	480	843	462
46	353	638	545	398	704	520	224	422	412	677	473	906	282

47	324	314	191	106	275	91	225	104	244	248	332	487	638
48	388	234	127	124	219	113	289	168	215	270	254	431	606
49	447	664	571	424	730	546	250	448	438	703	499	932	188
50	360	606	513	366	672	488	192	390	380	645	441	874	273
51	605	133	180	312	287	418	264	425	112	575	55	439	313
52	656	932	839	692	998	814	518	716	706	971	767	1200	444
53	573	113	101	280	79	329	359	403	163	402	157	235	509
54	293	384	274	143	319	129	262	60	314	286	402	531	675
55	372	283	199	153	229	39	273	151	324	196	324	441	686
56	330	479	386	239	545	361	65	259	253	518	314	747	378
57	610	297	234	391	107	275	511	389	322	248	331	254	693
58	598	874	781	634	940	756	460	658	648	913	709	1142	370
59	214	604	504	364	663	479	402	348	534	636	622	872	599
60	154	401	308	161	460	276	199	78	331	433	419	669	612
61	70	464	371	224	523	339	262	208	394	496	482	732	567
62	606	349	266	387	183	216	507	385	354	147	363	357	715
63	631	228	175	412	38	296	532	410	240	306	272	210	640
64	642	129	176	349	121	369	428	472	232	428	226	187	578
65	503	779	686	539	845	661	365	563	553	818	614	1047	245
66	372	762	662	522	821	637	456	506	644	794	705	1030	506
67	641	348	265	422	152	262	542	420	314	189	362	313	714
68	561	837	744	597	903	719	423	621	611	876	672	1105	311
69	478	754	661	514	820	636	340	538	528	793	589	1022	261
70	247	316	223	76	360	176	170	39	246	333	334	572	583
71	317	336	212	95	289	105	218	79	266	262	354	501	631
72	272	382	289	142	448	264	84	162	234	421	295	650	497
73	575	276	199	356	86	240	476	354	287	250	296	266	672
74	219	479	386	239	545	361	167	259	317	518	378	747	474
75	293	683	583	443	742	558	238	427	426	715	487	951	352
76	54	529	429	289	588	404	327	273	459	561	547	797	595
77	648	188	182	355	68	316	434	430	238	362	232	154	584
78	211	601	501	361	660	476	231	345	479	633	480	869	466
79	568	844	751	604	910	726	430	628	618	883	679	1112	348
80	497	150	67	204	70	257	288	327	155	335	164	282	516
81	290	680	580	440	739	555	317	424	505	712	566	948	506
82	475	65	42	182	137	282	261	295	65	439	85	321	437
83	654	341	278	435	151	319	555	433	366	266	375	298	755
84	445	387	276	226	294	50	346	224	364	125	410	506	759
85	375	765	665	525	824	640	384	509	572	797	633	1033	434
86	268	658	558	418	717	533	231	402	419	690	480	926	420
87	261	519	426	279	585	401	141	299	329	558	390	787	422
88	710	184	271	417	239	487	496	530	300	546	249	168	616
89	396	291	180	177	198	53	297	175	268	218	305	410	710
90	295	424	278	183	308	118	302	100	354	275	442	520	715
91	651	927	834	687	993	809	513	711	701	966	762	1195	407
92	175	565	472	325	624	440	311	309	495	597	583	833	504
93	585	67	146	292	135	385	371	405	175	458	147	249	499
94	250	640	540	400	699	515	277	384	465	672	526	908	466
95	717	221	251	424	137	385	503	499	307	417	301	95	653
96	246	454	361	213	373	183	332	130	384	340	472	585	745

97	788	302	322	495	208	456	574	570	378	488	372	23	724
98	426	596	503	356	662	478	182	380	370	635	431	864	253
99	596	575	330	377	237	179	497	375	552	100	589	407	941
100	634	910	817	670	976	792	496	694	684	949	745	1178	414
101	507	209	111	201	139	172	408	286	213	329	223	351	575
102	463	186	79	155	157	161	251	216	167	318	206	369	558
103	408	169	105	116	242	298	131	229	57	455	118	437	468
104	529	389	278	310	236	127	430	308	366	125	403	447	755
105	192	412	319	172	477	293	160	154	342	450	430	680	573
106	529	286	231	310	177	165	430	308	319	182	328	379	680
107	434	710	617	470	776	592	296	494	484	749	545	978	346
108	535	811	718	571	877	693	397	595	585	850	646	1079	336
109	630	108	191	337	165	424	416	450	220	483	188	190	540
110	446	252	145	227	162	111	347	225	233	268	272	374	624
111	166	518	425	277	437	247	336	114	448	404	536	649	749
112	471	313	202	252	166	99	372	250	290	210	358	378	679
113	442	718	625	478	784	600	304	502	492	757	553	986	300
114	523	230	147	304	81	188	424	302	235	255	174	293	596
115	566	45	139	273	228	383	352	386	121	540	60	314	411
116	235	313	220	73	371	187	168	43	243	344	331	583	581
117	432	822	722	582	881	697	441	566	629	854	690	1090	491
118	435	653	560	413	719	535	239	437	427	692	488	921	220
119	369	167	79	77	205	259	153	190	97	416	185	435	537
120	121	511	418	271	570	386	309	255	441	543	529	779	518

VECTOR													
64	**96**	**10**	**73**	**39**	**109**	**21**	**88**	**97**	**62**	**93**	**92**	**59**	**31**
DISTANCIAS													
484	**340**	**250**	**141**	**216**	**43**	**108**	**138**	**343**	**318**	**616**	**103**	**287**	**29**
SOLUCIÓN													
29261													

14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
257	187	91	412	400	472	389	610	340	510	153	511	269	525	150
641	577	450	624	752	805	665	76	730	152	447	844	283	157	539
541	477	357	531	659	712	572	183	630	134	354	751	190	95	446
407	337	210	384	512	565	425	317	490	217	207	604	42	232	299
706	636	509	690	818	871	731	192	789	248	470	910	332	42	598
522	452	325	506	634	687	547	442	605	370	280	726	148	257	414
184	375	248	210	338	391	251	396	394	175	246	430	169	316	279
391	321	141	408	536	589	449	430	474	330	113	628	63	355	283
372	507	380	398	526	579	439	202	582	46	377	618	213	160	469
679	609	482	663	791	844	704	515	762	527	437	883	305	330	571
433	595	468	459	587	640	500	141	643	72	465	679	301	167	557
915	845	718	892	1020	1073	933	233	998	438	715	1112	544	254	807
390	572	661	227	524	413	274	492	444	380	659	407	582	521	488
0	196	228	169	151	257	146	723	125	359	345	296	382	638	112
196	0	158	351	270	342	328	653	185	553	275	381	312	568	96
228	158	0	383	371	443	360	526	311	426	98	482	185	441	120
169	351	383	0	167	220	53	700	223	385	500	259	365	615	267
151	270	371	167	0	57	112	828	67	513	488	96	493	743	255
257	342	443	220	57	0	165	881	139	566	560	39	546	796	327
146	328	360	53	112	165	0	741	168	426	477	204	406	656	244
723	653	526	700	828	881	741	0	806	213	523	920	359	188	615
125	185	311	223	67	139	168	806	0	569	428	178	465	721	195
359	553	426	385	513	566	426	213	569	0	423	605	259	206	515
345	275	98	500	488	560	477	523	428	423	0	599	170	438	237
296	381	482	259	96	39	204	920	178	605	599	0	585	835	366
382	312	185	365	493	546	406	359	465	259	170	585	0	274	274
638	568	441	615	743	796	656	188	721	206	438	835	274	0	530
112	96	120	267	255	327	244	615	195	515	237	366	274	530	0
196	88	77	351	339	411	328	583	279	483	205	450	242	498	63
167	59	101	322	310	382	299	596	250	496	218	412	255	511	56
238	238	372	303	138	126	248	867	101	649	489	165	526	782	256
154	63	105	309	297	369	286	600	237	500	222	408	259	515	34
423	605	637	357	280	217	279	1018	336	703	754	194	683	933	521
372	302	175	338	466	519	379	357	455	257	172	558	27	272	264
618	548	421	602	730	783	643	538	701	466	376	822	244	353	509
566	496	369	550	678	731	591	449	649	377	324	770	192	264	458
740	670	504	724	852	905	765	360	823	346	504	944	366	179	632
229	382	250	255	350	436	296	351	439	130	248	475	136	271	286
823	753	626	807	935	988	848	272	906	365	587	1027	449	159	715
209	286	159	285	413	466	326	429	439	226	157	505	94	344	190
146	322	306	112	240	293	153	596	296	281	338	332	261	511	220
206	388	420	45	204	257	101	670	260	355	537	296	335	585	304
262	444	476	196	119	136	118	857	175	542	593	129	522	772	360
69	261	293	94	142	224	84	687	146	372	410	263	352	602	177
86	124	156	241	226	298	218	651	166	406	273	337	310	566	40
110	292	324	61	125	178	38	714	181	399	441	217	379	629	208

437	367	240	421	549	602	462	390	520	290	238	641	64	227	329
501	431	304	485	613	666	526	310	584	232	302	705	128	163	393
204	386	418	36	202	255	88	740	258	425	535	294	405	655	302
117	299	331	46	150	203	63	682	206	367	448	242	347	597	215
448	648	520	474	602	655	515	193	658	89	517	694	354	220	610
413	595	627	347	270	200	269	1008	326	693	744	177	673	923	511
686	616	489	663	791	844	704	131	769	209	486	883	322	57	578
451	381	201	458	586	639	499	460	534	360	151	678	101	318	343
485	415	288	469	597	650	510	413	568	370	241	689	111	228	377
119	276	260	150	278	331	191	555	334	240	297	370	220	470	174
723	653	526	707	835	888	748	302	806	368	487	927	349	149	615
355	537	569	289	212	197	211	950	268	635	686	157	615	865	453
223	49	185	378	319	391	355	680	234	580	302	430	339	595	123
298	228	63	453	441	513	430	477	381	377	47	552	136	392	190
191	121	27	346	334	406	323	540	274	440	162	445	199	455	83
719	649	522	703	831	884	744	375	802	400	483	923	345	194	611
744	674	547	728	856	909	769	233	827	286	508	948	370	80	636
755	685	558	732	860	913	773	98	838	278	555	952	391	132	647
260	442	474	141	151	168	116	855	207	540	591	162	520	770	358
209	209	343	285	120	108	230	838	72	631	460	147	497	753	227
754	684	557	738	866	919	779	344	837	360	518	958	380	193	646
318	500	532	252	175	192	174	913	231	598	649	186	578	828	416
235	417	449	116	132	149	91	830	188	515	566	159	495	745	333
360	290	163	366	494	547	407	392	443	292	133	586	37	307	252
430	360	233	414	542	595	455	412	513	312	213	634	53	248	397
180	315	188	280	408	461	321	458	464	221	186	500	123	373	193
688	618	491	672	800	853	713	289	771	333	452	892	314	127	580
83	172	149	253	235	339	230	555	228	304	263	378	220	470	79
50	232	264	131	101	174	108	759	105	413	381	213	418	674	148
219	92	82	374	362	434	351	605	302	505	227	473	264	520	119
761	691	564	738	866	919	779	177	844	284	561	958	390	100	653
74	81	182	243	189	261	220	677	129	406	299	300	336	592	105
325	507	539	259	182	150	181	920	238	605	656	127	585	835	423
610	540	413	587	715	768	628	216	693	201	410	807	246	28	502
139	98	261	285	155	227	229	756	88	492	378	266	415	671	144
588	518	391	565	693	746	606	141	671	111	388	785	224	95	480
767	697	570	751	879	932	792	346	850	412	531	971	393	193	659
558	488	361	542	670	723	583	478	641	406	316	762	184	293	450
160	245	346	213	48	59	158	841	42	559	463	98	500	756	230
53	138	239	199	132	204	202	734	72	406	356	243	393	649	123
43	200	232	211	183	294	178	595	162	316	342	333	260	510	98
823	753	626	800	928	981	841	108	906	321	623	1020	459	241	715
509	439	312	493	621	674	534	382	592	310	273	713	135	197	401
491	421	241	498	626	679	539	500	574	400	191	718	141	307	383
408	590	622	342	265	282	264	1003	321	688	739	276	668	918	506
128	76	146	283	271	343	260	641	211	541	263	382	300	556	32
698	628	501	675	803	856	716	57	781	221	498	895	334	131	590
99	89	221	245	178	250	220	716	96	452	338	289	375	631	105
830	760	633	807	935	988	848	190	913	353	631	1027	459	169	722
472	402	237	528	656	709	569	530	555	430	167	748	171	372	364

901	831	704	818	1006	1059	919	203	984	424	701	1098	530	240	793
183	365	397	27	181	234	82	672	237	357	514	273	337	587	281
709	639	512	693	821	874	734	421	792	460	467	913	335	236	601
391	573	605	325	248	185	247	986	304	671	722	162	651	901	489
620	550	423	604	732	785	645	275	703	259	384	824	202	87	512
576	506	379	553	681	734	594	262	659	213	376	773	176	115	468
315	451	325	341	469	522	382	245	525	72	322	561	158	200	413
642	572	445	626	754	807	667	420	725	408	400	846	268	235	534
228	235	108	383	371	443	360	488	311	388	167	482	153	403	119
642	572	445	626	754	807	667	344	725	365	406	846	268	159	534
191	373	405	125	82	142	47	786	145	471	522	145	451	701	289
292	474	506	226	94	76	129	887	154	572	623	75	552	802	390
743	673	546	720	848	901	761	43	826	266	543	940	379	161	635
559	489	362	543	671	724	584	346	642	279	323	763	185	161	451
435	365	150	590	578	650	567	594	518	494	90	689	235	436	402
584	514	387	568	696	749	609	350	667	332	348	788	210	165	476
199	381	413	81	137	184	53	794	209	479	530	194	459	709	297
636	566	439	620	748	801	661	265	719	281	400	840	262	75	528
679	609	482	656	784	837	697	81	762	132	479	876	315	189	571
348	278	151	364	492	545	405	389	431	289	137	584	48	304	240
217	302	403	270	105	69	215	898	99	616	520	108	557	813	287
192	374	406	29	190	243	64	729	246	414	523	282	394	644	290
482	412	286	459	587	640	500	243	555	111	283	679	119	163	375
142	99	84	297	285	357	274	587	225	487	209	396	246	502	35

66	**46**	**41**	**107**	**85**	**1**	**2**	**26**	**120**	**50**	**44**	**108**	**52**	**18**	**100**

243	**126**	**198**	**135**	**375**	**534**	**283**	**246**	**245**	**48**	**230**	**160**	**270**	**248**	**52**

29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
80	130	401	134	666	259	505	453	627	339	710	243	376	449	505
507	520	791	524	942	281	447	358	334	275	283	353	520	594	781
414	427	691	431	849	188	336	247	251	187	254	260	427	501	688
267	280	551	284	702	40	286	234	408	94	491	113	280	354	541
566	579	850	583	1008	364	354	265	168	313	117	419	586	660	847
382	395	666	399	824	180	110	59	239	281	375	235	402	476	663
305	318	474	314	528	142	406	354	528	45	611	89	106	180	367
251	264	535	268	726	72	284	232	406	184	489	133	300	378	565
437	450	662	454	716	211	424	335	300	143	319	283	294	368	555
539	552	823	556	981	337	70	182	166	438	354	392	559	633	820
525	538	723	542	777	299	461	372	348	204	351	368	355	429	616
775	788	1059	792	1210	549	566	477	331	543	202	621	788	862	1049
572	543	524	530	446	555	819	767	700	458	679	502	340	256	289
196	167	238	154	423	372	618	566	740	229	823	209	146	206	262
88	59	238	63	605	302	548	496	670	382	753	286	322	388	444
77	101	372	105	637	175	421	369	504	250	626	159	306	420	476
351	322	303	309	357	338	602	550	724	255	807	285	112	45	196
339	310	138	297	280	466	730	678	852	350	935	413	240	204	119
411	382	126	369	217	519	783	731	905	436	988	466	293	257	136
328	299	248	286	279	379	643	591	765	296	848	326	153	101	118
583	596	867	600	1018	357	538	449	360	351	272	429	596	670	857
279	250	101	237	336	455	701	649	823	439	906	439	296	260	175
483	496	649	500	703	257	466	377	346	130	365	226	281	355	542
205	218	489	222	754	172	376	324	504	248	587	157	338	537	593
450	412	165	408	194	558	822	770	944	475	1027	505	332	296	129
242	255	526	259	683	27	244	192	366	136	449	94	261	335	522
498	511	782	515	933	272	353	264	179	271	159	344	511	585	772
63	56	256	34	521	264	509	458	632	286	715	190	220	304	360
0	25	340	29	605	232	478	426	600	312	683	216	322	388	444
25	0	311	22	576	245	491	439	613	325	696	229	293	359	415
340	311	0	298	416	516	762	710	884	519	967	500	376	340	255
29	22	298	0	563	249	494	443	617	321	700	225	253	346	402
605	576	416	563	0	656	920	868	1042	573	1125	603	430	394	161
232	245	516	249	656	0	276	224	398	102	481	67	234	308	495
478	491	762	494	920	276	0	95	162	377	345	331	498	572	759
426	439	710	443	868	224	95	0	177	325	337	279	446	520	707
600	613	884	617	1042	398	162	177	0	425	183	453	620	694	881
312	325	519	321	573	102	377	325	425	0	430	96	151	225	412
683	696	967	700	1125	481	345	337	183	430	0	536	703	777	964
216	229	500	225	603	67	331	279	453	96	536	0	181	255	442
322	293	376	253	430	234	498	446	620	151	703	181	0	82	269
388	359	340	346	394	308	572	520	694	225	777	255	82	0	233
444	415	255	402	161	495	759	707	881	412	964	442	269	233	0
261	232	247	219	361	325	589	537	711	242	794	272	99	106	200
124	95	227	82	495	300	546	494	668	276	751	207	212	278	334
292	263	261	250	315	352	616	564	738	269	821	299	126	82	154

297	310	581	314	739	95	187	135	309	196	392	150	317	391	578
361	374	645	378	803	159	209	120	219	229	336	214	381	455	642
386	357	338	344	392	378	642	590	764	295	847	325	152	68	231
299	270	286	257	340	320	584	532	706	237	789	267	94	58	179
577	591	738	595	792	352	514	425	401	219	404	421	370	444	631
595	566	406	553	42	646	910	858	1032	563	1115	593	420	384	151
546	559	830	563	981	320	425	336	236	314	176	392	559	633	820
311	324	595	328	776	132	225	173	353	233	436	187	354	428	615
345	358	629	362	787	143	135	83	263	244	346	198	365	439	626
276	247	414	207	468	193	457	405	579	110	662	140	46	120	307
583	596	867	600	1025	381	239	231	77	408	106	436	603	677	864
537	508	348	495	84	588	852	800	974	505	1057	535	362	326	93
115	86	287	90	632	329	575	523	697	409	780	313	349	415	471
158	171	442	175	707	126	372	320	494	201	577	110	291	490	546
47	64	335	68	600	189	435	383	557	269	640	173	295	383	439
579	592	863	596	1021	377	139	154	22	447	209	432	599	673	860
604	617	888	621	1046	402	293	261	138	351	79	457	624	698	885
615	628	899	632	1050	389	465	376	260	383	161	461	628	702	889
442	413	287	400	201	493	757	705	879	410	962	440	267	231	44
311	282	29	269	398	487	733	681	855	501	938	471	354	322	237
614	627	898	631	1056	412	185	200	23	439	165	467	634	708	895
500	471	311	458	154	551	815	763	937	468	1020	498	325	289	65
417	388	268	375	226	468	732	680	854	385	937	415	242	206	55
220	233	504	237	684	34	272	220	394	146	477	95	262	336	523
290	378	574	382	732	88	201	149	323	189	406	143	310	384	571
245	258	544	228	598	96	360	308	482	91	565	29	142	250	437
548	561	832	565	990	346	237	205	82	373	141	401	568	642	829
139	134	289	112	507	193	457	405	579	174	662	126	154	290	346
232	203	206	190	385	408	654	602	776	283	859	248	140	168	224
31	43	363	58	628	254	500	448	622	334	705	238	327	411	467
621	634	905	638	1056	395	412	323	194	389	95	467	634	708	895
150	121	190	108	497	326	572	520	694	276	777	310	204	280	336
507	478	318	465	98	558	822	770	944	475	1027	505	332	296	63
470	483	754	487	905	244	353	264	184	243	187	316	483	557	744
176	164	140	136	424	405	651	599	773	362	856	389	290	322	263
448	461	732	465	883	222	378	289	272	216	254	294	461	535	722
627	640	911	644	1069	425	262	254	100	452	103	480	647	721	908
418	431	702	435	860	216	64	57	193	317	411	271	438	512	699
314	285	90	272	326	490	736	684	858	429	941	474	286	250	165
207	178	185	165	401	383	629	577	751	276	834	367	204	236	240
200	171	275	131	455	233	497	445	619	186	702	166	114	248	294
683	696	967	700	1118	457	583	494	378	451	279	529	696	770	957
369	382	653	386	811	167	157	67	215	268	315	222	389	463	650
351	364	635	368	816	172	214	162	342	273	425	227	394	468	655
590	561	401	548	174	641	905	853	1027	558	1110	588	415	379	155
76	47	272	30	537	290	536	484	658	318	741	222	256	320	376
558	571	842	575	993	332	481	392	303	326	215	404	571	645	832
189	160	151	147	447	365	611	559	733	322	816	349	250	282	286
690	703	974	707	1125	464	481	392	246	458	117	536	703	777	964
332	345	616	349	846	202	279	227	407	303	490	257	424	498	685

761	774	1045	778	1196	535	552	463	317	529	188	607	774	848	1035
365	336	317	323	371	310	574	522	696	227	779	257	84	19	210
569	582	853	586	1011	367	91	117	72	468	259	422	589	663	850
573	544	384	531	32	624	888	836	1010	541	1093	571	398	362	129
480	493	764	497	922	278	268	141	173	252	256	333	500	574	761
436	449	720	453	871	210	257	168	219	206	274	282	449	523	710
382	394	605	398	659	155	394	305	344	86	359	228	237	311	498
502	515	786	519	944	300	80	68	112	401	299	355	522	596	783
165	178	372	154	637	126	389	337	511	162	594	71	207	420	476
502	515	786	519	944	300	137	106	75	370	231	355	522	596	783
373	344	225	331	238	424	688	636	810	341	893	371	198	162	77
474	445	202	432	175	525	789	737	911	442	994	472	299	263	58
603	616	887	620	1038	877	520	431	315	371	216	449	616	690	877
419	432	703	436	861	217	207	141	162	272	279	272	439	513	700
295	383	579	387	844	189	343	291	471	295	554	247	428	627	683
444	457	728	461	886	242	156	61	135	311	276	297	464	538	725
381	352	285	339	261	432	696	644	818	349	901	379	206	126	90
496	509	780	513	938	294	284	195	104	321	193	349	516	590	777
539	552	823	556	974	313	479	390	366	307	308	385	552	626	813
208	221	492	225	682	32	283	231	405	144	488	93	260	334	521
371	342	77	329	383	547	793	741	915	486	998	531	343	307	222
374	345	326	332	380	367	631	579	753	284	836	314	141	78	219
343	356	626	360	777	116	355	266	305	108	322	189	356	429	616
29	42	286	36	551	236	482	430	604	316	687	220	268	334	390

58	**25**	**91**	**105**	**110**	**60**	**94**	**14**	**22**	**33**	**113**	**42**	**27**	**103**	**102**
157	**276**	**622**	**330**	**313**	**291**	**99**	**125**	**336**	**261**	**126**	**585**	**200**	**137**	**112**

44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
322	185	353	324	388	447	360	605	656	573	293	372	330	610	598
611	575	638	314	234	664	606	133	932	113	384	283	479	297	874
518	482	545	191	127	571	513	180	839	101	274	199	386	234	781
371	335	398	106	124	424	366	312	692	280	143	153	239	391	634
677	634	704	275	219	730	672	287	998	79	319	229	545	107	940
493	450	520	91	113	546	488	418	814	329	129	39	361	275	756
197	231	224	225	289	250	192	264	518	359	262	273	65	511	460
395	319	422	104	168	448	390	425	716	403	60	151	259	389	658
372	419	412	244	215	438	380	112	706	163	314	324	253	322	648
650	607	677	248	270	703	645	575	971	402	286	196	518	248	913
446	480	473	332	254	499	441	55	767	157	402	324	314	331	709
879	843	906	487	431	932	874	439	1200	235	531	441	747	254	1142
359	462	282	638	606	188	273	313	444	509	675	686	378	693	370
69	86	110	437	501	204	117	448	413	686	451	485	119	723	355
261	124	292	367	431	386	299	648	595	616	381	415	276	653	537
293	156	324	240	304	418	331	520	627	489	201	288	260	526	569
94	241	61	421	485	36	46	474	347	663	458	469	150	707	289
142	226	125	549	613	202	150	602	270	791	586	597	278	835	212
224	298	178	602	666	255	203	655	200	844	639	650	331	888	197
84	218	38	462	526	88	63	515	269	704	499	510	191	748	211
687	651	714	390	310	740	682	193	1008	131	460	413	555	302	950
146	166	181	520	584	258	206	658	326	769	534	568	334	806	268
372	406	399	290	232	425	367	89	693	209	360	370	240	368	635
410	273	441	238	302	535	448	517	744	486	151	241	297	487	686
263	337	217	641	705	294	242	694	177	883	678	689	370	927	157
352	310	379	64	128	405	347	354	673	322	101	111	220	349	615
602	566	629	227	163	655	597	220	923	57	318	228	470	149	865
177	40	208	329	393	302	215	610	511	578	343	377	174	615	453
261	124	292	297	361	386	299	577	595	546	311	345	276	583	537
232	95	263	310	374	357	270	591	566	559	324	358	247	596	508
247	227	261	581	645	338	286	738	406	830	595	629	414	867	348
219	82	250	314	378	344	257	595	553	563	328	362	207	600	495
361	495	315	739	803	392	340	792	42	981	776	787	468	1025	84
325	300	352	95	159	378	320	352	646	320	132	143	193	381	588
589	546	616	187	209	642	584	514	910	425	225	135	457	239	852
537	494	564	135	120	590	532	425	858	336	173	83	405	231	800
711	668	738	309	219	764	706	401	1032	236	353	263	579	77	974
242	276	269	196	229	295	237	219	563	314	233	244	110	408	505
794	751	821	392	336	847	789	404	1115	176	436	346	662	106	1057
272	207	299	150	214	325	267	421	593	392	187	198	140	436	535
99	212	126	317	381	152	94	370	420	559	354	365	46	603	362
106	278	82	391	455	68	58	444	384	633	428	439	120	677	326
200	334	154	578	642	231	179	631	151	820	615	626	307	864	93
0	151	46	408	472	130	48	461	351	650	445	456	126	694	293
151	0	182	365	429	276	189	495	485	614	379	413	166	651	427
46	182	0	435	499	96	31	488	305	677	472	483	164	721	247

408	365	435	0	64	461	403	385	729	353	83	54	276	292	671
472	429	499	64	0	525	467	307	793	220	147	72	340	236	735
130	276	96	461	525	0	82	514	382	603	498	509	190	747	324
48	189	31	403	467	82	0	456	330	645	440	451	132	689	272
461	495	488	385	307	514	456	0	782	210	455	377	329	384	724
351	485	305	729	793	382	330	782	0	971	766	777	458	1015	85
650	614	677	353	220	603	645	210	971	0	375	300	518	186	913
445	379	472	83	147	498	440	455	766	375	0	90	313	336	708
456	413	483	54	72	509	451	377	777	300	90	0	324	246	719
126	166	164	276	340	190	132	329	458	518	313	324	0	562	400
694	651	721	292	236	747	689	384	1015	186	336	246	562	0	957
293	427	247	671	735	324	272	724	85	913	708	719	400	957	0
288	151	319	394	458	413	326	675	622	643	408	442	303	680	564
363	226	396	191	255	488	401	471	697	440	138	239	250	477	639
256	119	287	254	378	381	294	534	590	503	268	302	249	540	532
690	647	717	288	232	743	685	416	1011	262	332	242	558	103	953
715	672	742	313	257	768	710	325	1036	117	357	267	583	69	978
719	683	746	422	330	772	714	279	1040	75	430	340	587	191	982
198	332	152	576	640	174	177	629	199	818	613	624	305	862	125
218	198	243	552	616	320	268	720	388	801	566	600	396	838	330
725	682	752	323	233	778	720	415	1046	231	367	277	593	59	988
256	390	210	634	698	287	235	687	152	876	671	682	363	920	77
173	307	127	551	615	149	152	604	224	793	588	599	280	837	150
353	288	380	92	156	406	348	387	674	355	83	139	221	377	616
401	358	428	21	85	454	396	407	722	375	62	68	269	306	664
267	159	294	179	243	320	262	310	588	421	216	227	96	465	530
659	616	686	257	201	712	654	349	980	165	301	211	527	35	922
136	621	194	276	340	288	201	393	497	518	313	324	108	562	439
41	122	72	473	537	166	89	502	375	722	487	521	167	759	317
284	147	315	319	383	409	322	600	618	568	333	367	281	605	560
725	689	752	333	277	778	720	285	1046	81	377	287	593	125	988
143	37	184	391	455	278	191	495	487	640	405	439	166	677	429
263	397	217	641	705	294	242	694	96	883	678	689	370	927	30
574	538	601	199	135	627	569	217	895	85	292	228	442	167	837
195	116	226	470	534	320	243	581	414	719	484	518	252	756	356
552	516	579	255	169	605	547	138	873	92	325	241	420	255	815
738	695	765	336	280	791	733	428	1059	230	380	290	606	44	1001
529	486	556	127	149	582	524	454	850	365	165	75	397	311	792
169	201	171	555	619	248	196	648	316	804	569	603	324	841	258
109	86	140	448	512	234	157	495	391	697	462	496	166	734	333
67	90	142	316	380	246	159	405	445	558	353	364	76	602	387
787	751	814	490	448	840	782	310	1108	184	560	458	655	309	1050
480	437	507	78	53	533	475	358	801	269	122	32	348	215	743
485	419	512	114	151	538	480	495	806	364	40	79	353	325	748
346	480	300	724	788	377	325	777	173	966	761	772	453	1010	90
193	56	224	355	419	318	231	636	527	604	369	403	210	641	469
662	626	689	365	261	715	657	200	983	74	435	356	530	245	925
155	76	186	430	494	280	203	541	437	679	444	478	212	716	379
794	758	821	402	346	847	789	354	1115	150	446	356	662	169	1057
515	400	542	157	221	568	510	525	836	429	70	144	383	390	778

865	829	892	473	417	918	860	425	1186	221	517	427	733	240	1128
83	255	67	393	457	65	35	446	361	635	430	441	122	679	303
680	637	707	278	221	733	675	642	1001	308	316	212	548	153	943
329	463	283	707	771	360	308	760	33	949	744	755	436	983	52
591	548	618	138	74	644	586	276	912	144	233	143	459	156	854
540	504	567	112	48	593	535	259	861	172	195	120	408	202	803
328	362	355	188	160	381	323	169	649	208	259	269	196	327	591
613	570	640	211	169	666	608	456	934	307	249	225	481	193	876
232	136	324	209	273	418	331	482	627	451	246	256	161	494	569
613	570	640	211	155	666	608	381	934	231	255	165	481	125	876
129	263	83	507	571	160	108	560	228	749	544	555	236	793	170
230	364	184	608	672	261	209	661	160	850	645	656	337	894	140
707	671	734	410	306	760	702	241	1028	104	480	395	575	246	970
530	487	557	128	50	583	525	325	851	233	172	82	398	179	793
500	363	531	221	285	625	538	589	834	493	134	208	372	454	776
555	512	582	153	93	608	550	380	876	237	197	107	423	170	818
137	271	91	515	579	111	116	568	259	757	552	563	244	801	185
607	564	634	205	149	660	602	297	928	147	249	159	475	87	870
643	607	670	346	266	696	638	112	964	158	416	342	511	335	906
351	276	378	103	167	404	346	384	672	352	95	150	219	388	614
226	258	228	612	676	305	253	705	373	861	626	660	381	898	315
123	264	84	450	514	34	75	503	370	692	487	498	179	736	312
446	411	473	149	121	499	441	238	767	206	220	230	315	288	709
207	70	238	301	365	332	245	581	541	550	315	349	222	587	483

47	**114**	**80**	**16**	**45**	**98**	**43**	**79**	**19**	**65**	**81**	**118**	**6**	**90**	**34**

205	**80**	**413**	**156**	**255**	**210**	**63**	**150**	**168**	**295**	**308**	**535**	**118**	**172**	**88**

59	60	61	62	63	64	65	66	67	68	69	70	71	72	73
214	154	70	606	631	642	503	372	641	561	478	247	317	272	575
604	401	464	349	228	129	779	762	348	837	754	316	336	382	276
504	308	371	266	175	176	686	662	265	744	661	223	212	289	199
364	161	224	387	412	349	539	522	422	597	514	76	95	142	356
663	460	523	183	38	121	845	821	152	903	820	360	289	448	86
479	276	339	216	296	369	661	637	262	719	636	176	105	264	240
402	199	262	507	532	428	365	456	542	423	340	170	218	84	476
348	78	208	385	410	472	563	506	420	621	538	39	79	162	354
534	331	394	354	240	232	553	644	314	611	528	246	266	234	287
636	433	496	147	306	428	818	794	189	876	793	333	262	421	250
622	419	482	363	272	226	614	705	362	672	589	334	354	295	296
872	669	732	357	210	187	1047	1030	313	1105	1022	572	501	650	266
599	612	567	715	640	578	245	506	714	311	261	583	631	497	672
223	298	191	719	744	755	260	209	754	318	235	360	430	180	688
49	228	121	649	674	685	442	209	684	500	417	290	360	315	618
185	63	27	522	547	558	474	343	557	532	449	163	233	188	491
378	453	346	703	728	732	141	285	738	252	116	366	414	280	672
319	441	334	831	856	860	151	120	866	175	132	494	542	408	800
391	513	406	884	909	913	168	108	919	192	149	547	595	461	853
355	430	323	744	769	773	116	230	779	174	91	407	455	321	713
680	477	540	375	233	98	855	838	344	913	830	392	412	458	289
234	381	274	802	827	838	207	72	837	231	188	443	513	464	771
580	377	440	400	286	278	540	631	360	598	515	292	312	221	333
302	47	162	483	508	555	591	460	518	649	566	133	213	186	452
430	552	445	923	948	952	162	147	958	186	159	586	634	500	892
339	136	199	345	370	391	520	497	380	578	495	37	53	123	314
595	392	455	194	80	132	770	753	193	828	745	307	248	373	127
123	190	83	611	636	647	358	227	646	416	333	252	397	193	580
115	158	47	579	604	615	442	311	614	500	417	220	290	245	548
86	171	64	592	617	628	413	282	627	471	388	233	378	258	561
287	442	335	863	888	899	287	29	898	311	268	504	574	544	832
90	175	68	596	621	632	400	269	631	458	375	237	382	228	565
632	707	600	1021	1046	1050	201	398	1056	154	226	684	732	598	990
329	126	189	377	402	389	493	487	412	551	468	34	88	96	346
575	372	435	139	293	465	757	733	185	815	732	272	201	360	237
523	320	383	154	261	376	705	681	200	763	680	220	149	308	205
697	494	557	22	138	260	879	855	23	937	854	394	323	482	82
409	201	269	447	351	383	410	501	439	468	385	146	189	91	373
780	577	640	209	79	161	962	938	165	1020	937	477	406	565	141
313	110	173	432	457	461	440	471	467	498	415	95	143	29	401
349	291	295	599	624	628	267	354	634	325	242	262	310	142	568
415	490	383	673	698	702	231	322	708	289	206	336	384	250	642
471	546	439	860	885	889	44	237	895	65	55	523	571	437	829
288	363	256	690	715	719	198	218	725	256	173	353	401	267	659
151	226	119	647	672	683	332	198	682	390	307	288	358	159	616
319	396	287	717	742	746	152	243	752	210	127	380	428	294	686

394	191	254	288	313	422	576	552	323	634	551	92	21	179	257
458	255	378	232	257	330	640	616	233	698	615	156	85	243	201
413	488	381	743	768	772	174	320	778	287	149	406	454	320	712
326	401	294	685	710	714	177	268	720	235	152	348	396	262	654
675	471	534	416	325	279	629	720	415	687	604	387	407	310	349
622	697	590	1011	1036	1040	199	388	1046	152	224	674	722	588	980
643	440	503	262	117	75	818	801	231	876	793	355	375	421	165
408	138	268	332	357	430	613	566	367	671	588	83	62	216	301
442	239	302	242	267	340	624	600	277	682	599	139	68	227	211
303	250	249	558	583	587	305	396	593	363	280	221	269	96	527
680	477	540	103	69	191	862	838	59	920	837	377	306	465	35
564	639	532	953	978	982	125	330	988	77	150	616	664	530	922
0	255	148	676	701	712	469	258	711	527	444	317	387	342	645
255	0	115	473	498	509	544	413	508	602	519	92	184	139	442
148	115	0	536	561	572	437	306	571	495	412	177	247	209	505
676	473	536	0	153	275	858	834	45	916	833	373	302	461	97
701	498	561	153	0	122	883	859	122	941	858	398	327	486	56
712	509	572	275	122	0	887	870	244	945	862	424	444	490	178
469	544	437	858	883	887	0	269	893	66	25	521	569	435	827
258	413	306	834	859	870	269	0	869	293	250	475	545	526	803
711	508	571	45	122	244	893	869	0	951	868	408	337	496	66
527	602	495	916	941	945	66	293	951	0	91	579	627	493	885
444	519	412	833	858	862	25	250	868	91	0	496	544	410	802
317	92	177	373	398	424	521	475	408	579	496	0	70	124	342
387	184	247	302	327	444	569	545	337	627	544	70	0	172	271
342	139	209	461	486	490	435	526	496	493	410	124	172	0	430
645	442	505	97	56	178	827	803	66	885	802	342	271	430	0
199	216	153	558	583	587	344	260	593	402	319	221	269	97	527
259	334	227	755	780	791	222	177	790	280	197	396	466	219	724
84	180	53	601	626	637	465	334	636	523	440	242	312	267	570
718	515	578	209	56	66	893	876	178	951	868	418	347	496	112
160	252	145	673	698	709	334	161	708	392	309	314	384	196	642
534	609	502	923	948	952	103	300	958	56	120	586	634	500	892
567	364	427	199	108	160	742	725	198	800	717	279	220	345	132
147	331	224	752	777	788	295	111	787	319	276	393	463	275	721
545	342	405	287	175	161	720	703	286	778	695	257	277	323	220
724	521	584	122	113	235	906	882	77	964	881	421	350	509	79
515	312	375	170	332	405	697	673	216	755	672	212	141	300	276
294	416	309	837	862	873	197	72	872	221	178	478	548	454	832
187	309	202	730	755	766	272	156	765	296	253	371	441	227	699
227	295	195	598	623	627	292	246	633	350	267	261	309	137	567
780	577	640	393	240	118	955	938	362	1013	930	492	512	558	296
466	263	326	206	236	309	648	624	238	706	623	163	92	251	180
448	178	308	321	346	419	653	606	356	711	628	123	93	256	290
617	692	585	1006	1031	1035	162	383	1041	96	187	669	717	583	975
103	216	109	637	662	673	374	243	672	432	349	278	423	225	606
655	452	515	318	176	62	830	813	287	888	805	367	387	433	232
138	291	184	712	737	748	318	122	747	342	299	353	423	235	681
787	584	647	272	125	92	962	945	228	1020	937	487	416	565	181
429	174	289	386	411	484	683	587	421	741	658	153	132	286	355

858	655	718	343	196	173	1033	1016	299	1091	1008	558	487	636	252
392	467	360	675	700	704	208	299	710	266	183	338	386	252	644
666	463	526	47	212	334	848	824	95	906	823	363	292	451	156
600	675	568	989	1014	1018	169	366	1024	122	194	652	700	566	958
577	374	437	208	177	219	759	735	187	817	734	274	159	362	121
533	330	393	254	195	247	708	691	233	766	683	204	133	311	167
478	276	339	391	280	277	496	587	358	554	471	191	211	177	292
599	396	459	89	231	353	781	757	135	839	756	296	225	384	175
262	120	110	490	515	590	474	343	525	532	449	127	202	74	459
599	396	459	77	155	247	781	757	89	839	756	296	225	384	99
400	475	368	789	814	818	76	207	824	133	51	452	500	366	758
501	576	469	890	915	919	91	184	925	113	88	553	601	467	859
700	497	560	348	177	55	875	858	299	933	850	412	432	478	233
516	313	376	175	200	273	698	674	176	756	673	213	142	301	144
392	137	252	450	475	548	681	550	485	739	656	150	196	276	419
541	338	401	147	191	277	723	699	149	781	698	238	167	326	135
408	483	376	797	822	826	60	267	832	126	35	460	508	374	766
593	390	453	119	108	192	775	751	118	833	750	290	219	378	52
636	433	496	381	266	155	811	794	380	869	786	348	368	414	314
305	94	165	384	409	421	519	463	419	577	494	12	92	122	353
351	473	366	894	919	930	254	66	929	278	235	535	605	511	863
401	476	369	732	757	761	137	308	767	275	112	395	443	309	701
439	237	300	352	243	275	614	587	319	672	589	152	172	218	253
126	162	55	583	608	619	388	257	618	446	363	224	294	249	552

71	**32**	**117**	**76**	**116**	**83**	**99**	**23**	**119**	**40**	**38**	**61**	**111**	**78**	**77**
382	**329**	**394**	**230**	**432**	**172**	**460**	**111**	**189**	**96**	**269**	**252**	**389**	**715**	**287**

74	75	76	77	78	79	80	81	82	83	84	85	86	87	88
219	293	54	648	211	568	497	290	475	654	445	375	268	261	710
479	683	529	188	601	844	150	680	65	341	387	765	658	519	184
386	583	429	182	501	751	67	580	42	278	276	665	558	426	271
239	443	289	355	361	604	204	440	182	435	226	525	418	279	417
545	742	588	68	660	910	70	739	137	151	294	824	717	585	239
361	558	404	316	476	726	257	555	282	319	50	640	533	401	487
167	238	327	434	231	430	288	317	261	555	346	384	231	141	496
259	427	273	430	345	628	327	424	295	433	224	509	402	299	530
317	426	459	238	479	618	155	505	65	366	364	572	419	329	300
518	715	561	362	633	883	335	712	439	266	125	797	690	558	546
378	487	547	232	480	679	164	566	85	375	410	633	480	390	249
747	951	797	154	869	1112	282	948	321	298	506	1033	926	787	168
474	352	595	584	466	348	516	506	437	755	759	434	420	422	616
83	50	219	761	74	325	610	139	588	767	558	160	53	43	823
172	232	92	691	81	507	540	98	518	697	488	245	138	200	753
149	264	82	564	182	539	413	261	391	570	361	346	239	232	626
253	131	374	738	243	259	587	285	565	751	542	213	199	211	800
235	101	362	866	189	182	715	155	693	879	670	48	132	183	928
339	174	434	919	261	150	768	227	746	932	723	59	204	294	981
230	108	351	779	220	181	628	229	606	792	583	158	202	178	841
555	759	605	177	677	920	216	756	141	346	478	841	734	595	108
228	105	302	844	129	238	693	88	671	850	641	42	72	162	906
304	413	505	284	406	605	201	492	111	412	406	559	406	316	321
263	381	227	561	299	656	410	378	388	531	316	463	356	342	623
378	213	473	958	300	127	807	266	785	971	762	98	243	333	1020
220	418	264	390	336	585	246	415	224	393	184	500	393	260	459
470	674	520	100	592	835	28	671	95	193	293	756	649	510	241
79	148	119	653	105	423	502	144	480	659	450	230	123	98	715
139	232	31	621	150	507	470	176	448	627	418	314	207	200	683
134	203	43	634	121	478	483	164	461	640	431	285	178	171	696
289	206	363	905	190	318	754	140	732	911	702	90	185	275	967
112	190	58	638	108	465	487	136	465	644	435	272	165	131	700
507	385	628	1056	497	98	905	424	883	1069	860	326	401	455	1118
193	408	254	395	326	558	244	405	222	425	216	490	383	233	457
457	654	500	412	572	822	353	651	378	262	64	736	629	497	583
405	602	448	323	520	770	264	599	289	254	57	684	577	445	494
579	776	622	194	694	944	184	773	272	100	193	858	751	619	378
174	283	334	389	276	475	243	362	216	452	317	429	276	186	451
662	859	705	95	777	1027	187	856	254	103	411	941	834	702	279
126	248	238	467	310	505	316	389	294	480	271	474	367	166	529
154	140	327	634	204	332	483	290	461	647	438	286	204	114	696
290	168	411	708	280	296	557	322	535	721	512	250	236	248	770
346	224	467	895	336	63	744	263	722	908	699	165	240	294	957
136	41	284	725	143	263	574	195	552	738	529	169	109	67	787
621	122	147	689	37	397	538	116	516	695	486	201	86	90	751
194	72	315	752	184	217	601	226	579	765	556	171	140	142	814

276	473	319	333	391	641	199	470	255	336	127	555	448	316	490
340	537	383	277	455	705	135	534	169	280	149	619	512	380	448
288	166	409	778	278	294	627	320	605	791	582	248	234	246	840
201	89	322	720	191	242	569	243	547	733	524	196	157	159	782
393	502	600	285	495	694	217	581	138	428	454	648	495	405	310
497	375	618	1046	487	96	895	414	873	1059	850	316	391	445	1108
518	722	568	81	640	883	85	719	92	230	365	804	697	558	184
313	487	333	377	405	678	292	484	325	380	165	569	462	353	560
324	521	367	287	439	689	228	518	241	290	75	603	496	364	458
108	167	281	593	166	370	442	252	420	606	397	324	166	76	655
562	759	605	125	677	927	167	756	255	44	311	841	734	602	309
439	317	560	988	429	30	837	356	815	1001	792	258	333	387	1050
199	259	84	718	160	534	567	147	545	724	515	294	187	227	780
216	334	180	515	252	609	364	331	342	521	312	416	309	295	577
153	227	53	578	145	502	427	224	405	584	375	309	202	195	640
558	755	601	209	673	923	199	752	287	122	170	837	730	598	393
583	780	626	56	698	948	108	777	175	113	332	862	755	623	240
587	791	637	66	709	952	160	788	161	235	405	873	766	627	118
344	222	465	893	334	103	742	295	720	906	697	197	272	292	955
260	177	334	876	161	300	725	111	703	882	673	72	156	246	938
593	790	636	178	708	958	198	787	286	77	216	872	765	633	362
402	280	523	951	392	56	800	319	778	964	755	221	296	350	1013
319	197	440	868	309	120	717	276	695	881	672	178	253	267	930
221	396	242	418	314	586	279	393	257	421	212	478	371	261	492
269	466	312	347	384	634	220	463	277	350	141	548	441	309	512
97	219	267	496	196	500	345	275	323	509	300	454	227	137	558
527	724	570	112	642	892	132	721	220	79	276	832	699	567	296
0	134	170	593	99	409	442	178	420	606	397	263	130	69	655
134	0	255	797	125	287	646	154	624	803	594	128	68	82	859
170	255	0	643	173	530	492	190	470	649	440	337	230	223	705
593	797	643	0	715	958	128	794	167	169	352	879	772	633	179
99	125	173	715	0	399	564	79	542	721	512	164	57	90	777
409	287	530	958	399	0	807	326	785	971	762	228	303	357	1020
442	646	492	128	564	807	0	643	88	211	293	728	627	482	269
178	154	190	794	79	326	643	0	621	800	591	130	86	176	856
420	624	470	167	542	785	88	621	0	299	318	706	599	460	229
606	803	649	169	721	971	211	800	299	0	355	885	778	646	353
397	594	440	352	512	762	293	591	318	355	0	676	569	437	523
263	128	337	879	164	228	728	130	706	885	676	0	107	197	941
130	68	230	772	57	303	627	86	599	778	569	107	0	90	834
69	82	223	633	90	357	482	176	460	646	437	197	90	0	695
655	859	705	179	777	1020	269	856	229	353	523	941	834	695	0
348	545	391	256	463	713	197	542	222	259	97	627	520	388	427
353	527	373	366	445	718	281	524	365	369	154	609	502	393	537
492	370	613	1041	482	112	890	409	868	1054	845	311	386	440	1103
104	164	99	679	57	439	528	112	506	685	476	246	114	134	741
530	734	580	120	652	895	159	731	104	289	421	816	709	570	121
138	114	212	754	39	349	603	40	581	760	551	138	46	136	816
662	866	712	69	784	1027	197	863	236	213	421	948	841	702	162
383	508	354	431	426	748	346	505	395	434	219	590	483	423	630

733	937	783	154	855	1098	268	934	307	284	492	1019	912	773	138
267	145	388	710	257	273	559	299	537	723	514	227	213	225	772
548	745	591	268	663	913	241	742	506	172	133	827	720	588	452
475	353	596	1024	465	66	873	392	851	1037	828	294	369	423	1086
459	656	502	197	574	824	59	653	147	200	208	738	631	499	328
408	612	458	215	530	773	87	609	121	246	197	694	587	448	350
260	369	403	283	362	561	172	448	110	371	334	515	362	272	345
481	678	524	287	596	846	240	675	320	212	84	760	653	521	471
96	264	187	526	182	539	375	261	353	538	329	346	239	165	588
481	678	524	211	596	846	164	675	252	148	168	760	653	521	365
275	153	396	824	265	140	673	233	651	837	628	135	210	223	886
376	176	497	925	276	110	774	242	752	938	729	123	219	324	987
575	779	625	121	697	940	189	776	149	290	460	861	754	615	80
398	595	441	220	513	763	108	592	187	223	147	677	570	438	391
353	471	317	495	389	746	410	468	459	498	283	553	446	432	694
423	620	466	224	538	788	139	617	244	214	118	702	595	462	395
283	161	404	832	273	155	681	293	659	845	636	195	270	231	894
475	672	518	139	590	840	80	669	168	131	224	754	647	515	310
511	715	561	213	633	876	182	712	97	379	419	797	690	551	189
219	384	230	429	302	584	276	381	254	432	223	466	359	259	489
320	185	394	936	221	285	785	177	763	942	733	57	164	254	998
276	154	397	767	266	282	616	308	594	780	571	236	222	234	829
315	518	364	281	436	679	135	515	108	332	295	600	493	355	343
104	178	60	625	96	453	474	175	452	631	422	260	153	146	687

55	**24**	**53**	**82**	**115**	**5**	**3**	**67**	**37**	**112**	**54**	**35**	**101**	**4**	**106**
241	**486**	**92**	**97**	**228**	**137**	**265**	**23**	**135**	**197**	**225**	**268**	**201**	**310**	**155**

89	90	91	92	93	94	95	96	97	98	99	100	101	102	103
396	295	651	175	585	250	717	246	788	426	596	634	507	463	408
291	424	927	565	67	640	221	454	302	596	575	910	209	186	169
180	278	834	472	146	540	251	361	322	503	330	817	111	79	105
177	183	687	325	292	400	424	213	495	356	377	670	201	155	116
198	308	993	624	135	699	137	373	208	662	237	976	139	157	242
53	118	809	440	385	515	385	183	456	478	179	792	172	161	298
297	302	513	311	371	277	503	332	574	182	497	496	408	251	131
175	100	711	309	405	384	499	130	570	380	375	694	286	216	229
268	354	701	495	175	465	307	384	378	370	552	684	213	167	57
218	275	966	597	458	672	417	340	488	635	100	949	329	318	455
305	442	762	583	147	526	301	472	372	431	589	745	223	206	118
410	520	1195	833	249	908	95	585	23	864	407	1178	351	369	437
710	715	407	504	499	466	653	745	724	253	941	414	575	558	468
509	491	408	128	698	99	830	472	901	183	709	391	620	576	315
439	421	590	76	628	89	760	402	831	365	639	573	550	506	451
312	241	622	146	501	221	633	237	704	397	512	605	423	379	325
493	498	342	283	675	245	807	528	818	27	693	325	604	553	341
621	626	265	271	803	178	935	656	1006	181	821	248	732	681	469
674	679	282	343	856	250	988	709	1059	234	874	185	785	734	522
534	539	264	260	716	220	848	569	919	82	734	247	645	594	382
382	500	1003	641	57	716	190	530	203	672	421	986	275	262	245
592	574	321	211	781	96	913	555	984	237	792	304	703	659	525
310	400	688	541	221	452	353	430	424	357	460	671	259	213	72
273	191	739	263	498	338	631	167	701	514	467	722	384	376	322
713	718	276	382	895	289	1027	748	1098	273	913	162	824	773	561
135	141	668	300	334	375	459	171	530	337	335	651	202	176	158
197	307	918	556	131	631	169	372	240	587	236	901	87	115	200
401	383	506	32	590	105	722	364	793	281	601	489	512	468	413
369	351	590	76	558	189	690	332	761	365	569	573	480	436	382
382	364	561	47	571	160	703	345	774	336	582	544	493	449	394
653	635	401	272	842	151	974	616	1045	317	853	384	764	720	605
386	368	548	30	575	147	707	349	778	323	586	531	497	453	398
811	816	174	537	993	447	1125	846	1196	371	1011	32	922	871	659
167	172	641	290	332	365	464	202	535	310	367	624	278	210	155
157	214	905	536	481	611	481	279	552	574	91	888	268	257	394
67	162	853	484	392	559	392	227	463	522	117	836	141	168	305
215	342	1027	658	303	733	246	407	317	696	72	1010	173	219	344
268	273	558	318	326	322	458	303	529	227	468	541	252	206	86
315	425	1110	741	215	816	117	490	188	779	259	1093	256	274	359
222	227	588	222	404	349	536	257	607	257	422	571	333	282	228
389	394	415	256	571	250	703	424	774	84	589	398	500	449	237
463	468	379	320	645	282	777	498	848	19	663	362	574	523	311
650	655	155	376	832	286	964	685	1035	210	850	129	761	710	498
480	485	346	193	662	155	794	515	865	83	680	329	591	540	328
437	419	480	56	626	76	758	400	829	255	637	463	548	504	362
507	512	300	224	689	186	821	542	892	67	707	283	618	567	355

78	114	724	355	365	430	402	157	473	393	278	707	138	112	188
53	151	788	419	261	494	346	221	417	457	221	771	74	48	160
533	538	377	318	715	280	847	568	918	65	733	360	644	593	381
475	480	325	231	657	203	789	510	860	35	675	308	586	535	323
358	495	777	636	200	541	354	525	425	446	642	760	276	259	169
801	806	173	527	983	437	1115	836	1186	361	1001	33	912	861	649
269	364	966	604	74	679	150	429	221	635	308	949	144	172	208
122	40	761	369	435	444	446	70	517	430	316	744	233	195	259
32	79	772	403	356	478	356	144	427	441	212	755	143	120	269
348	353	453	210	530	212	662	383	733	122	548	436	459	408	196
215	325	1010	641	245	716	169	390	240	679	153	983	156	202	327
743	748	90	469	925	379	1057	778	1128	303	943	52	854	803	591
466	448	617	103	655	138	787	429	858	392	666	600	577	533	478
263	178	692	216	452	291	584	174	655	467	463	675	374	330	276
326	308	585	109	515	184	647	289	718	360	526	568	437	393	339
206	321	1006	637	318	712	272	386	343	675	47	989	208	254	391
236	346	1031	662	176	737	125	411	196	700	212	1014	177	195	280
309	419	1035	673	62	748	92	484	173	704	334	1018	219	247	277
648	653	162	374	830	318	962	683	1033	208	848	169	759	708	496
624	606	383	243	813	122	945	587	1016	299	824	366	735	691	587
238	356	1041	672	287	747	228	421	299	710	95	1024	187	233	358
706	711	96	432	888	342	1020	741	1091	266	906	122	817	766	554
623	628	187	349	805	299	937	658	1008	183	823	194	734	683	471
163	123	669	278	367	353	487	153	558	338	363	652	274	204	191
92	93	717	423	387	423	416	132	487	386	292	700	159	133	211
251	256	583	225	433	235	565	286	636	252	451	566	362	311	177
180	290	975	606	232	681	181	355	252	644	156	958	121	167	292
348	353	492	104	530	138	662	383	733	267	548	475	459	408	260
545	527	370	164	734	114	866	508	937	145	745	353	656	612	369
391	373	613	99	580	212	712	354	783	388	591	596	502	458	403
256	366	1041	679	120	754	69	431	154	710	268	1024	197	215	283
463	445	482	57	652	39	784	426	855	257	663	465	574	530	362
713	718	112	439	895	349	1027	748	1098	273	913	66	824	773	561
197	281	890	528	159	603	197	346	268	559	241	873	59	87	172
542	524	409	112	731	40	863	505	934	299	742	392	653	609	448
222	365	868	506	104	581	236	395	307	537	506	851	147	121	110
259	369	1054	685	289	760	213	434	284	723	172	1037	200	246	371
97	154	845	476	421	551	421	219	492	514	133	828	208	197	334
627	609	311	246	816	138	948	590	1019	227	827	294	738	694	515
520	502	386	114	709	46	841	483	912	213	720	369	631	587	362
388	393	440	134	570	136	702	423	773	225	588	423	499	448	272
427	537	1103	741	121	816	162	630	138	772	452	1086	328	350	345
0	111	796	427	325	502	325	176	396	465	185	779	112	101	238
111	0	801	409	438	484	435	65	506	470	305	784	222	199	299
796	801	0	522	978	432	1110	831	1181	355	996	142	907	856	644
427	409	522	0	616	96	748	390	819	297	627	505	538	494	439
325	438	978	616	0	691	154	505	235	647	364	961	218	225	220
502	484	432	96	691	0	823	465	894	259	702	415	613	569	408
325	435	1110	748	154	823	0	500	81	779	322	1003	266	284	352
176	65	831	390	505	465	500	0	571	500	370	814	287	269	329

396	506	1181	819	235	894	81	571	0	850	393	1164	337	355	423
465	470	355	297	647	259	779	500	850	0	665	339	576	525	313
185	305	996	627	364	702	322	370	393	665	0	979	208	251	388
779	784	142	505	961	415	1003	814	1164	339	979	0	890	839	627
112	222	907	538	218	613	266	287	337	576	208	890	0	46	183
101	199	856	494	225	569	284	269	355	525	251	839	46	0	137
238	299	644	439	220	408	352	329	423	313	388	627	183	137	0
133	238	929	560	363	635	362	303	433	598	52	912	156	199	336
280	286	622	151	463	221	595	294	666	397	480	605	391	341	287
137	244	929	560	287	635	294	309	365	598	80	912	131	177	314
579	584	223	305	761	241	893	614	964	139	779	206	690	639	427
680	685	205	406	862	265	994	715	1065	240	880	143	791	740	528
364	474	1023	661	41	736	147	550	160	692	394	1006	248	270	265
51	161	846	477	289	552	289	226	360	515	167	829	49	66	203
240	129	829	353	569	428	564	80	635	604	434	812	351	333	393
67	186	871	502	293	577	293	251	364	540	128	854	80	123	260
587	592	222	313	769	301	901	622	972	108	787	229	698	647	435
128	238	923	554	208	629	208	303	279	592	161	906	69	115	240
323	456	959	597	93	672	247	486	284	628	607	942	244	218	178
174	135	667	266	364	341	498	165	569	336	374	650	285	215	188
684	666	368	303	873	195	1005	647	1076	284	884	351	795	751	572
522	527	365	306	704	268	836	557	907	56	722	348	633	582	370
199	260	762	401	218	475	350	290	421	431	349	745	144	98	39
373	355	536	47	562	135	694	336	765	311	573	519	484	440	386

63	**12**	**72**	**8**	**89**	**70**	**56**	**13**	**95**	**11**	**9**	**49**	**75**	**74**	**7**

210	**650**	**162**	**175**	**163**	**221**	**378**	**653**	**301**	**61**	**438**	**166**	**134**	**167**	**231**

104	105	106	107	108	109	110	111	112	113	114	115	116	117	118
529	192	529	434	535	630	446	166	471	442	523	566	235	432	435
389	412	286	710	811	108	252	518	313	718	230	45	313	822	653
278	319	231	617	718	191	145	425	202	625	147	139	220	722	560
310	172	310	470	571	337	227	277	252	478	304	273	73	582	413
236	477	177	776	877	165	162	437	166	784	81	228	371	881	719
127	293	165	592	693	424	111	247	99	600	188	383	187	697	535
430	160	430	296	397	416	347	336	372	304	424	352	168	441	239
308	154	308	494	595	450	225	114	250	502	302	386	43	566	437
366	342	319	484	585	220	233	448	290	492	235	121	243	629	427
125	450	182	749	850	483	268	404	210	757	255	540	344	854	692
403	430	328	545	646	188	272	536	358	553	174	60	331	690	488
447	680	379	978	1079	190	374	649	378	986	293	314	583	1090	921
755	573	680	346	336	540	624	749	679	300	596	411	581	491	220
642	228	642	191	292	743	559	435	584	199	636	679	348	217	192
572	235	572	373	474	673	489	365	514	381	566	609	278	302	374
445	108	445	405	506	546	362	150	387	413	439	482	151	403	406
626	383	626	125	226	720	543	590	568	81	620	656	364	270	29
754	371	754	82	94	848	671	578	696	137	748	784	492	105	190
807	443	807	142	76	901	724	650	749	184	801	837	545	69	243
667	360	667	47	129	761	584	567	609	53	661	697	405	215	64
420	488	344	786	887	43	346	594	350	794	265	81	389	898	729
725	311	725	145	154	826	642	518	667	209	719	762	431	99	246
408	388	365	471	572	266	279	494	332	479	281	132	289	616	414
400	167	406	522	623	543	323	90	348	530	400	479	137	520	523
846	482	846	145	75	940	763	689	788	194	840	876	584	108	282
268	153	268	451	552	379	185	235	210	459	262	315	48	557	394
235	403	159	701	802	161	161	436	165	709	75	189	304	813	644
534	119	534	289	390	635	451	402	476	297	528	571	240	287	290
502	165	502	373	474	603	419	295	444	381	496	539	208	371	374
515	178	515	344	445	616	432	383	457	352	509	552	221	342	345
786	372	786	225	202	887	703	579	728	285	780	823	492	77	326
519	154	519	331	432	620	436	387	461	339	513	556	225	329	332
944	637	944	238	175	1038	861	844	886	261	938	974	682	383	380
300	126	300	424	525	877	217	189	242	432	294	313	32	547	367
80	389	137	688	789	520	207	343	156	696	284	479	283	793	631
68	337	106	636	737	431	141	291	61	644	195	390	231	741	579
112	511	75	810	911	315	162	471	135	818	104	366	405	915	753
401	162	370	341	442	371	272	295	311	349	321	307	144	486	284
299	594	231	893	994	216	279	554	276	901	193	308	488	998	836
355	71	355	371	472	449	272	247	297	379	349	385	93	531	314
522	207	522	198	299	616	439	428	464	206	516	552	260	343	141
596	420	596	162	263	690	513	627	538	126	590	626	334	307	78
783	476	783	77	58	877	700	683	725	90	777	813	521	222	219
613	232	613	129	230	707	530	500	555	137	607	643	351	226	123
570	136	570	263	364	671	487	363	512	271	564	607	276	258	264
640	324	640	83	184	734	557	531	582	91	634	670	378	228	84

211	209	211	507	608	410	128	221	153	515	205	346	103	612	450
169	273	155	571	672	306	50	285	93	579	149	266	167	676	514
666	418	666	160	261	760	583	625	608	111	660	696	404	305	34
608	331	608	108	209	702	525	538	550	116	602	638	346	253	75
456	482	381	560	661	241	325	589	380	568	297	112	384	705	503
934	627	934	228	160	1028	851	834	876	259	928	964	672	373	370
307	451	231	749	850	104	233	493	237	757	147	158	352	861	692
249	246	255	544	645	480	172	134	197	552	249	416	95	626	487
225	256	165	555	656	395	82	208	107	563	159	342	150	660	498
481	161	481	236	337	575	398	372	423	244	475	511	219	381	179
193	494	125	793	894	246	179	454	170	801	87	335	388	898	736
876	569	876	170	140	970	793	776	818	185	870	906	614	315	312
599	262	599	400	501	700	516	392	541	408	593	636	305	351	401
396	120	396	475	576	497	313	137	338	483	390	433	94	473	476
459	110	459	368	469	560	376	252	401	376	453	496	165	366	369
89	490	77	789	890	348	175	450	147	797	119	381	384	894	732
231	515	155	814	915	177	200	475	191	822	108	266	409	919	757
353	590	247	818	919	55	273	548	277	826	192	155	421	930	761
781	474	781	76	91	875	698	681	723	60	775	811	519	254	137
757	343	757	207	184	858	674	550	699	267	751	794	463	66	308
135	525	89	824	925	299	176	485	149	832	118	380	419	929	767
839	532	839	133	113	933	756	739	781	126	833	869	577	278	275
756	449	756	51	88	850	673	656	698	35	750	786	494	235	112
296	127	296	452	553	412	213	150	238	460	290	348	12	535	395
225	202	225	500	601	432	142	196	167	508	219	368	92	605	443
384	74	384	366	467	478	301	276	326	374	378	414	122	511	309
175	459	99	758	859	233	144	419	135	766	52	314	353	863	701
481	96	481	275	376	575	398	353	423	283	475	511	219	320	276
678	264	678	153	176	779	595	471	620	161	672	715	384	185	154
524	187	524	396	497	625	441	317	466	404	518	561	230	394	397
287	526	211	824	925	121	220	495	224	832	139	213	429	936	767
596	182	596	265	276	697	513	389	538	273	590	633	302	221	266
846	539	846	140	110	940	763	746	788	155	840	876	584	285	282
240	375	164	673	774	189	108	410	139	681	80	182	276	785	616
675	261	675	233	242	776	592	468	617	293	669	712	381	177	308
320	353	252	651	752	149	187	459	244	659	168	97	254	763	594
212	538	148	837	938	290	223	498	214	845	131	379	432	942	780
84	329	168	628	729	460	147	283	118	636	224	419	223	733	571
760	346	760	135	123	861	677	553	702	195	754	797	466	57	236
653	239	653	210	219	754	570	446	595	270	647	690	359	164	222
521	165	521	223	324	615	438	432	462	231	515	551	259	254	234
471	588	365	886	987	80	391	694	395	894	310	189	489	998	829
133	280	137	579	680	364	51	240	67	587	128	323	174	684	522
238	286	244	584	685	474	161	129	186	592	238	456	135	666	527
929	622	929	223	205	1023	846	829	871	222	923	959	667	368	365
560	151	560	305	406	661	477	353	502	313	554	597	266	303	306
363	463	287	761	862	41	289	569	293	769	208	93	364	873	704
635	221	635	241	265	736	552	428	577	301	629	672	341	195	268
362	595	294	893	994	147	289	564	293	901	208	247	498	1005	836
303	294	309	614	715	550	226	80	251	622	303	486	165	647	557

433	666	365	964	1065	160	360	635	364	972	279	284	569	1076	907
598	397	598	139	240	692	515	604	540	108	592	628	336	284	56
52	480	80	779	880	394	167	434	128	787	161	607	374	884	722
912	605	912	206	143	1006	829	812	854	229	906	942	650	351	348
156	391	131	690	791	248	49	351	80	698	69	244	285	795	633
199	341	177	639	740	270	66	333	123	647	115	218	215	751	582
336	287	314	427	528	265	203	393	260	435	240	178	188	572	370
0	413	87	712	813	393	115	367	76	720	160	421	307	817	655
413	0	413	405	506	508	330	257	355	413	407	444	115	403	406
87	413	0	712	813	317	98	373	70	720	84	346	307	817	655
712	405	712	0	82	806	629	612	654	55	706	742	450	192	148
813	506	813	82	0	907	730	713	755	123	807	843	551	145	249
393	508	317	806	907	0	319	614	323	814	238	124	409	819	749
115	330	98	629	730	319	0	290	39	637	92	284	224	734	572
367	257	373	612	713	614	290	0	315	620	367	550	154	610	613
76	355	70	654	755	323	39	315	0	662	87	345	249	759	597
720	413	720	55	123	814	637	620	662	0	714	750	458	252	77
160	407	84	706	807	238	92	367	87	714	0	262	301	811	649
421	444	346	742	843	124	284	550	345	750	262	0	345	854	685
307	115	307	450	551	409	224	154	249	458	301	345	0	523	393
817	403	817	192	145	819	734	610	759	252	811	854	523	0	293
655	406	655	148	249	749	572	613	597	77	649	685	393	293	0
297	248	275	545	646	263	164	354	221	553	201	199	149	657	488
506	169	506	319	520	607	423	299	448	327	500	543	212	317	320

86	**68**	**69**	**20**	**17**	**84**	**36**	**57**	**104**	**48**	**29**	**15**	**30**	**87**	**28**
296	**91**	**91**	**53**	**542**	**57**	**231**	**193**	**169**	**361**	**88**	**59**	**171**	**98**	**610**

119	120
369	121
167	511
79	418
77	271
205	570
259	386
153	309
190	255
97	441
416	543
185	529
435	779
537	518
482	142
412	99
286	84
459	297
587	285
640	357
500	274
243	587
555	225
111	487
283	209
679	396
119	246
163	502
375	35
343	29
356	42
626	286
360	36
777	551
116	236
355	482
266	430
305	604
108	316
322	687
189	220
356	268
429	334
616	390
446	207
411	70
473	238

149 301
121 365
499 332
441 245
238 581
767 541
206 550
220 315
230 349
315 222
288 587
709 483
439 126
237 162
300 55
352 583
243 608
275 619
614 388
587 257
319 618
672 446
589 363
152 224
172 294
218 249
253 552
315 104
518 178
364 60
281 625
436 96
679 453
135 474
515 175
108 452
332 631
295 422
600 260
493 153
355 146
343 687
199 373
260 355
762 536
401 47
218 562
475 135
350 694
290 336

421	765
431	311
349	573
745	519
144	484
98	440
39	386
297	506
248	169
275	506
545	319
646	520
263	607
164	423
354	299
221	448
553	327
201	500
199	543
149	212
657	317
488	320
0	347
347	0

51	64

279

SOLUCIÓN DEL HK48 USANDO INDICE Y SOLVER

48

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	273	1272	744	1138	1972	1580	1878	1539	1457	429	1129	1251
2	273	0	999	809	866	1722	1338	1640	1226	1185	440	894	992
3	1272	999	0	1519	140	937	697	951	267	227	1229	587	369
4	744	809	1519	0	1425	1861	1473	1713	1761	1617	370	1073	1304
5	1138	866	140	1425	0	1052	776	1049	402	361	1119	578	406
6	1972	1722	937	1861	1052	0	400	182	820	721	1735	851	740
7	1580	1338	697	1473	776	400	0	304	699	538	1335	454	393
8	1878	1640	951	1713	1049	182	304	0	884	755	1612	749	690
9	1539	1226	267	1761	402	820	699	884	0	177	1486	757	506
10	1457	1185	227	1617	361	721	538	755	177	0	1362	587	335
11	429	440	1229	370	1119	1735	1335	1612	1486	1362	0	891	1082
12	1129	894	587	1073	578	851	454	749	757	587	891	0	252
13	1251	992	369	1304	406	740	393	690	506	335	1082	252	0
14	1421	1173	554	1369	618	551	173	476	609	435	1199	308	222
15	588	334	721	1092	581	1551	1198	1501	981	930	726	803	814
16	334	358	1212	453	1095	1769	1370	1654	1474	1358	96	920	1094
17	837	626	739	798	670	1159	760	1049	967	819	583	309	510
18	1364	1124	596	1283	641	613	216	516	681	504	1125	238	235
19	229	358	1291	973	1152	2072	1692	1995	1552	1496	653	1252	1335
20	961	847	1114	565	1060	1300	919	1149	1317	1153	563	569	820
21	754	533	701	1315	567	1605	1286	1580	936	927	947	940	892
22	1169	915	426	1204	443	807	435	739	594	428	986	165	100
23	1488	1219	285	1796	374	1017	879	1079	197	341	1493	863	626
24	720	481	676	846	579	1251	861	1161	928	803	560	414	541
25	1280	1009	155	1447	235	818	548	815	316	180	1183	454	219
26	816	543	456	1143	325	1259	913	1214	723	649	813	552	524
27	664	937	1936	959	1802	2596	2198	2485	2203	2119	882	1745	1897
28	1178	915	319	1275	331	826	483	780	500	343	1033	269	90
29	939	667	337	1213	217	1137	803	1100	604	521	902	482	410
30	1698	1441	604	2085	665	1255	1181	1347	482	652	1763	1188	952
31	983	812	907	742	862	1123	731	985	1104	939	642	355	805
32	1119	848	214	1309	182	943	627	916	455	340	1032	397	238
33	1029	776	424	1479	312	1359	1086	1361	630	649	1131	833	706
34	1815	1560	748	1760	864	188	292	260	641	533	1604	713	570
35	721	526	817	703	732	1282	883	1171	1058	918	463	432	622
36	1753	1494	666	1727	783	271	279	328	562	451	1556	666	503
37	330	598	1592	872	1456	2300	1906	2202	1857	1783	663	1453	1581
38	1499	1244	521	1479	608	483	178	445	528	362	1298	410	257
39	1107	1304	2172	686	2066	2540	2156	2385	2425	2290	947	1758	1985
40	1576	1306	356	1698	491	609	490	665	220	130	1461	642	396
41	942	685	467	1057	400	1038	662	966	704	568	795	262	309
42	484	668	1583	387	1466	2099	1699	1969	1845	1727	371	1260	1453
43	617	444	882	1252	744	1776	1430	1729	1122	1105	882	1051	1039
44	896	1157	2139	904	2013	2699	2300	2568	2405	2301	967	1858	2043

45	1184	1359	2182	668	2082	2493	2117	2333	2428	2285	973	1737	1972
46	1030	1176	1961	443	1865	2266	1888	2108	2204	2059	768	1508	1744
47	1718	1475	781	1600	875	264	138	177	738	595	1472	592	514
48	604	335	678	930	552	1398	1023	1327	945	853	588	598	661

VECTOR

4 11 42 37 16 20 22 13 28 29 35 2 1 19

DISTANCIAS

370 371 504 586 642 725 100 90 320 529 526 273 229 676

SOLUCIÓN

16829

14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1421	588	334	837	1364	229	961	754	1169	1488	720	1280	816	664	1178
1173	334	358	626	1124	358	847	533	915	1219	481	1009	543	937	915
554	721	1212	739	596	1291	1114	701	426	285	676	155	456	1936	319
1369	1092	453	798	1283	973	565	1315	1204	1796	846	1447	1143	959	1275
618	581	1095	670	641	1152	1060	567	443	374	579	235	325	1802	331
551	1551	1769	1159	613	2072	1300	1605	807	1017	1251	818	1259	2596	826
173	1198	1370	760	216	1692	919	1286	435	879	861	548	913	2198	483
476	1501	1654	1049	516	1995	1149	1580	739	1079	1161	815	1214	2485	780
609	981	1474	967	681	1552	1317	936	594	197	928	316	723	2203	500
435	930	1358	819	504	1496	1153	927	428	341	803	180	649	2119	343
1199	726	96	583	1125	653	563	947	986	1493	560	1183	813	882	1033
308	803	920	309	238	1252	569	940	165	863	414	454	552	1745	269
222	814	1094	510	235	1335	820	892	100	626	541	219	524	1897	90
0	1025	1227	617	90	1525	835	1114	263	770	700	400	740	2049	311
1025	0	663	632	999	572	972	225	763	908	451	767	293	1240	726
1227	663	0	610	1156	557	642	879	1000	1467	558	1178	780	831	1038
617	632	610	0	546	983	397	821	411	1023	180	651	478	1438	476
90	999	1156	546	0	1479	745	1105	240	831	645	442	723	1983	316
1525	572	557	983	1479	0	1163	676	1264	1473	839	1326	847	801	1254
835	972	642	397	745	1163	0	1183	725	1399	549	1004	869	1427	818
1114	225	879	821	1105	676	1183	0	865	821	644	790	388	1374	803
263	763	1000	411	240	1264	725	865	0	699	453	290	483	1809	107
770	908	1467	1023	831	1473	1399	821	699	0	950	410	690	2147	594
700	451	558	180	645	839	549	644	453	950	0	624	325	1356	480
400	767	1178	651	442	1326	1004	790	290	410	624	0	479	1941	188
740	293	780	478	723	847	869	388	483	690	325	479	0	1480	435
2049	1240	831	1438	1983	801	1427	1374	1809	2147	1356	1941	1480	0	1829
311	726	1038	476	316	1254	818	803	107	594	480	188	435	1829	0
630	420	879	485	623	976	882	484	384	590	369	350	129	1603	320
1087	1111	1726	1333	1152	1643	1716	968	1024	326	1241	736	949	2339	919
630	862	700	235	543	1157	214	1056	511	1191	413	792	708	1524	605
459	617	1023	525	470	1169	902	655	251	499	473	161	325	1780	154
924	443	1082	827	939	983	1222	318	712	504	680	547	355	1673	623
405	1374	1631	1022	482	1905	1210	1420	646	838	1097	632	1081	2421	652
739	586	488	123	669	878	390	794	525	1098	166	745	492	1315	580
360	1299	1579	973	443	1836	1184	1341	585	758	1038	552	1007	2394	582
1749	887	586	1155	1690	346	1225	1017	1499	1794	1049	1607	1137	357	1508
115	1070	1320	715	205	1590	949	1137	330	703	781	375	779	2136	344
2055	1633	982	1475	1969	1286	1239	1836	1885	2439	1497	2115	1759	825	1950
428	1057	1463	902	510	1621	1210	1056	495	414	905	296	774	2237	429
492	547	796	273	455	1034	660	679	231	751	238	392	291	1589	242
1568	999	371	953	1492	689	863	1200	1356	1837	925	1547	1148	579	1402
1256	252	802	882	1238	503	1207	189	999	1011	702	959	516	1204	949
2166	1483	940	1550	2091	995	1446	1645	1949	2374	1506	2121	1688	347	1986

2026	1681	1021	1467	1938	1376	1197	1891	1872	2455	1506	2114	1785	959	1943
1796	1489	826	1240	1709	1239	969	1704	1644	2237	1287	1890	1573	940	1717
303	1326	1508	898	354	1828	1042	1403	567	928	998	641	1038	2336	603
853	236	550	396	813	674	741	442	591	921	216	676	231	1266	582
21	**43**	**15**	**48**	**24**	**41**	**17**	**31**	**12**	**18**	**14**	**10**	**32**	**25**	**3**
189	**252**	**236**	**216**	**238**	**273**	**235**	**355**	**238**	**90**	**435**	**340**	**161**	**155**	**140**

29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
939	1698	983	1119	1029	1815	721	1753	330	1499	1107	1576	942	484	617
667	1441	812	848	776	1560	526	1494	598	1244	1304	1306	685	668	444
337	604	907	214	424	748	817	666	1592	521	2172	356	467	1583	882
1213	2085	742	1309	1479	1760	703	1727	872	1479	686	1698	1057	387	1252
217	665	862	182	312	864	732	783	1456	608	2066	491	400	1466	744
1137	1255	1123	943	1359	188	1282	271	2300	483	2540	609	1038	2099	1776
803	1181	731	627	1086	292	883	279	1906	178	2156	490	662	1699	1430
1100	1347	985	916	1361	260	1171	328	2202	445	2385	665	966	1969	1729
604	482	1104	455	630	641	1058	562	1857	528	2425	220	704	1845	1122
521	652	939	340	649	533	918	451	1783	362	2290	130	568	1727	1105
902	1763	642	1032	1131	1604	463	1556	663	1298	947	1461	795	371	882
482	1188	355	397	833	713	432	666	1453	410	1758	642	262	1260	1051
410	952	805	238	706	570	622	503	1581	257	1985	396	309	1453	1039
630	1087	630	459	924	405	739	360	1749	115	2055	428	492	1568	1256
420	1111	862	617	443	1374	586	1299	887	1070	1633	1057	547	999	252
879	1726	700	1023	1082	1631	488	1579	586	1320	982	1463	796	371	802
485	1333	235	525	827	1022	123	973	1155	715	1475	902	273	953	882
623	1152	543	470	939	482	669	443	1690	205	1969	510	455	1492	1238
976	1643	1157	1169	983	1905	878	1836	346	1590	1286	1621	1034	689	503
882	1716	214	902	1222	1210	390	1184	1225	949	1239	1210	660	863	1207
484	968	1056	655	318	1420	794	1341	1017	1137	1836	1056	679	1200	189
384	1024	511	251	712	646	525	585	1499	330	1885	495	231	1356	999
590	326	1191	499	504	838	1098	758	1794	703	2439	414	751	1837	1011
369	1241	413	473	680	1097	166	1038	1049	781	1497	905	238	925	702
350	736	792	161	547	632	745	552	1607	375	2115	296	392	1547	959
129	949	708	325	355	1081	492	1007	1137	779	1759	774	291	1148	516
1603	2339	1524	1780	1673	2421	1315	2394	357	2136	825	2237	1589	579	1204
320	919	605	154	623	652	580	582	1508	344	1950	429	242	1402	949
0	872	699	197	358	957	529	881	1263	660	1849	645	240	1250	631
872	0	1511	815	669	1092	1397	1019	1982	1010	2708	695	1061	2089	1148
699	1511	0	697	1051	1018	290	985	1280	743	1427	996	466	987	1110
197	815	697	0	469	761	607	685	1446	472	1969	457	254	1393	823
358	669	1051	469	0	1171	847	1089	1316	919	2063	776	598	1434	507
957	1092	1018	761	1171	0	1144	83	2145	317	2445	426	875	1972	1584
529	1397	290	607	847	1144	0	1094	1036	836	1371	1008	354	833	828
881	1019	985	685	1089	83	1094	0	2083	259	2412	345	811	1925	1507
1263	1982	1280	1446	1316	2145	1036	2083	0	1828	1005	1903	1272	504	849
660	1010	743	472	919	317	836	259	1828	0	2165	330	559	1668	1291
1849	2708	1427	1969	2063	2445	1371	2412	1005	2165	0	2377	1723	636	1720
645	695	996	457	776	426	1008	345	1903	330	2377	0	667	1829	1235
240	1061	466	254	598	875	354	811	1272	559	1723	667	0	1162	792
1250	2089	987	1393	1434	1972	833	1925	504	1668	636	1829	1162	0	1087
631	1148	1110	823	507	1584	828	1507	849	1291	1720	1235	792	1087	0
1802	2594	1584	1963	1926	2571	1429	2523	653	2264	534	2410	1744	600	1490

1867	2734	1395	1975	2101	2408	1369	2380	1114	2138	145	2367	1724	701	1787
1650	2520	1166	1752	1898	2179	1146	2151	1019	1908	290	2139	1500	550	1614
923	1212	861	739	1187	194	1021	220	2044	268	2281	519	796	1835	1553
341	1176	626	515	548	1231	352	1163	932	917	1531	972	361	917	486
5	**30**	**23**	**9**	**36**	**34**	**6**	**8**	**47**	**7**	**38**	**40**	**33**	**26**	**27**
665	**326**	**197**	**562**	**83**	**188**	**182**	**177**	**138**	**178**	**330**	**776**	**355**	**1480**	**347**

44	45	46	47	48
896	1184	1030	1718	604
1157	1359	1176	1475	335
2139	2182	1961	781	678
904	668	443	1600	930
2013	2082	1865	875	552
2699	2493	2266	264	1398
2300	2117	1888	138	1023
2568	2333	2108	177	1327
2405	2428	2204	738	945
2301	2285	2059	595	853
967	973	768	1472	588
1858	1737	1508	592	598
2043	1972	1744	514	661
2166	2026	1796	303	853
1483	1681	1489	1326	236
940	1021	826	1508	550
1550	1467	1240	898	396
2091	1938	1709	354	813
995	1376	1239	1828	674
1446	1197	969	1042	741
1645	1891	1704	1403	442
1949	1872	1644	567	591
2374	2455	2237	928	921
1506	1506	1287	998	216
2121	2114	1890	641	676
1688	1785	1573	1038	231
347	959	940	2336	1266
1986	1943	1717	603	582
1802	1867	1650	923	341
2594	2734	2520	1212	1176
1584	1395	1166	861	626
1963	1975	1752	739	515
1926	2101	1898	1187	548
2571	2408	2179	194	1231
1429	1369	1146	1021	352
2523	2380	2151	220	1163
653	1114	1019	2044	932
2264	2138	1908	268	917
534	145	290	2281	1531
2410	2367	2139	519	972
1744	1724	1500	796	361
600	701	550	1835	917
1490	1787	1614	1553	486
0	678	727	2435	1461

678	0	229	2238	1560
727	229	0	2010	1353
2435	2238	2010	0	1157
1461	1560	1353	1157	0

44	**39**	**45**	**46**	**4**

534	**145**	**229**	**443**	

SOLUCIÓN DEL SI175 USANDO INDICE Y SOLVER

175

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	113	189	299	189	177	187	187	187	162	187	162	187
2	113	0	177	291	201	189	187	187	187	162	187	162	187
3	189	177	0	248	255	245	234	224	213	202	189	177	187
4	299	291	248	0	342	335	328	321	314	306	299	291	284
5	189	201	255	342	0	113	134	149	167	177	189	201	213
6	177	189	245	335	113	0	127	134	149	167	177	189	201
7	187	187	234	328	134	127	0	113	134	149	167	177	189
8	187	187	224	321	149	134	113	0	113	134	149	167	177
9	187	187	213	314	167	149	134	113	0	127	134	149	167
10	162	162	202	306	177	167	149	134	127	0	127	134	149
11	187	187	189	299	189	177	167	149	134	127	0	127	134
12	162	162	177	291	201	189	177	167	149	134	127	0	127
13	187	187	187	284	213	201	189	177	167	149	134	127	0
14	162	162	162	277	223	213	201	189	177	167	149	134	127
15	187	187	187	269	234	223	213	202	189	177	167	149	134
16	177	167	162	259	245	234	223	213	202	189	177	167	149
17	189	187	187	248	255	245	234	224	213	202	189	177	167
18	202	189	162	238	266	255	245	234	224	213	202	189	177
19	213	202	187	227	274	266	255	245	234	224	213	202	189
20	234	224	167	205	289	281	274	266	256	245	234	224	213
21	245	234	177	194	296	289	281	274	266	256	245	234	224
22	256	245	189	187	304	296	289	282	274	266	256	245	234
23	266	256	202	170	311	304	296	289	282	274	266	256	245
24	274	266	213	187	319	311	304	296	289	282	274	266	256
25	282	274	224	162	326	319	311	304	296	289	282	274	266
26	289	282	234	187	333	326	319	311	304	296	289	282	274
27	296	289	245	162	340	333	326	319	311	304	296	289	282
28	311	304	266	187	354	347	340	333	326	319	311	304	296
29	319	311	274	162	361	354	347	340	333	326	319	311	304
30	326	319	282	187	367	361	354	347	340	333	326	319	311
31	333	326	289	187	373	367	361	354	347	340	333	326	319
32	340	333	296	185	379	373	367	361	354	347	340	333	326
33	262	262	262	342	220	220	196	196	196	220	196	220	213
34	246	246	246	335	202	202	177	177	177	202	177	202	201
35	262	262	262	328	220	220	196	196	196	220	196	220	196
36	246	246	246	326	202	202	177	177	177	202	177	202	185
37	262	262	262	321	220	220	196	196	196	220	196	220	196
38	312	312	312	315	280	280	263	263	263	280	263	280	263
39	262	262	262	314	220	220	196	196	196	220	196	220	196
40	262	262	262	306	220	220	196	196	196	220	196	220	196
41	262	262	262	304	220	220	196	196	196	220	196	220	196
42	278	278	278	299	241	241	218	218	218	241	218	241	218
43	278	278	278	293	241	241	218	218	218	241	218	241	218
44	278	278	278	291	241	241	218	218	218	241	218	241	218

45	278	278	278	284	241	241	218	218	218	241	218	241	218
46	278	278	278	283	241	241	218	218	218	241	218	241	218
47	278	278	278	278	241	241	218	218	218	241	218	241	218
48	278	278	278	278	241	241	218	218	218	241	218	241	218
49	262	262	262	269	234	223	213	202	196	220	196	220	196
50	246	246	246	259	245	234	223	213	202	202	177	202	177
51	246	246	246	258	246	236	225	214	203	202	178	202	177
52	262	262	262	262	255	245	234	224	213	220	196	220	196
53	246	246	246	246	262	251	241	230	219	208	197	202	177
54	262	262	262	262	266	255	245	234	224	220	202	220	196
55	246	246	246	246	274	266	255	245	234	224	213	202	189
56	262	262	262	262	281	274	266	256	245	234	224	220	202
57	246	246	246	246	284	277	269	259	248	238	227	216	205
58	262	262	262	262	289	281	274	266	256	245	234	224	213
59	246	246	246	246	295	287	280	273	264	253	243	232	221
60	262	262	262	262	296	289	281	274	266	256	245	234	224
61	256	246	246	246	304	296	289	282	274	266	256	245	234
62	262	262	262	262	306	298	291	284	276	269	258	248	237
63	266	256	202	187	311	304	296	289	282	274	266	256	245
64	272	263	246	246	317	309	302	294	287	280	272	263	253
65	274	266	213	187	319	311	304	296	289	282	274	266	256
66	282	274	262	262	326	319	311	304	296	289	282	274	266
67	283	276	246	246	327	320	313	305	298	290	283	276	268
68	289	282	262	262	333	326	319	311	304	296	289	282	274
69	294	286	278	278	337	330	323	316	309	301	294	286	279
70	296	289	278	278	340	333	326	319	311	304	296	289	282
71	304	296	278	278	347	340	333	326	319	311	304	296	289
72	305	297	278	278	348	341	334	327	320	312	305	297	290
73	311	304	278	278	354	347	340	333	326	319	311	304	296
74	316	308	278	278	358	351	344	337	330	323	316	308	301
75	319	311	278	278	361	354	347	340	333	326	319	311	304
76	326	319	282	278	367	361	354	347	340	333	326	319	311
77	333	326	289	278	373	367	361	354	347	340	333	326	319
78	314	314	314	348	283	283	266	266	266	283	266	283	266
79	294	294	294	332	262	262	240	240	240	262	240	262	240
80	320	320	320	321	289	289	272	272	272	289	272	289	272
81	320	320	320	320	289	289	272	272	272	289	272	289	272
82	278	278	278	299	241	241	218	218	218	241	218	241	218
83	273	273	273	288	234	234	211	211	211	234	211	234	211
84	287	287	287	287	253	253	230	230	230	253	230	253	230
85	320	320	320	320	289	289	272	272	272	289	272	289	272
86	287	287	287	287	255	253	234	230	230	253	230	253	230
87	273	273	273	273	270	261	250	240	229	234	211	234	211
88	287	287	287	287	288	280	273	264	254	253	232	253	230
89	320	320	320	320	299	291	284	277	272	289	272	289	272
90	314	314	314	314	310	303	295	288	280	283	266	283	266
91	294	294	294	294	321	314	306	299	291	284	277	269	259
92	312	312	312	312	332	325	318	310	303	295	288	280	273

93	314	314	314	314	342	335	328	321	314	306	299	291	284
94	314	314	314	314	353	346	339	332	325	318	310	303	295
95	321	314	314	314	363	356	349	342	335	328	321	314	306
96	345	345	345	348	316	316	300	300	300	316	300	316	300
97	294	294	294	310	262	262	240	240	240	262	240	262	240
98	314	314	314	314	283	283	266	266	266	283	266	283	266
99	314	314	314	314	283	283	266	266	266	283	266	283	266
100	294	294	294	294	262	262	240	240	240	262	240	262	240
101	312	312	312	312	280	280	263	263	263	280	263	280	263
102	294	294	294	294	270	262	250	240	240	262	240	262	240
103	314	314	314	314	288	283	273	266	266	283	266	283	266
104	314	314	314	314	299	291	284	277	269	283	266	283	266
105	312	312	312	312	321	314	306	299	291	284	277	280	263
106	294	294	294	294	332	325	318	310	303	295	288	280	273
107	314	314	314	314	342	335	328	321	314	306	299	291	284
108	314	314	314	314	353	346	339	332	325	318	310	303	295
109	321	314	314	314	363	356	349	342	335	328	321	314	306
110	345	345	345	345	334	327	320	313	305	316	300	316	300
111	345	345	345	345	364	357	350	343	336	329	322	316	308
112	370	370	370	370	344	344	329	329	329	344	329	344	329
113	370	370	370	370	344	344	329	329	329	344	329	344	329
114	370	370	370	370	344	344	329	329	329	344	329	344	329
115	370	370	370	370	344	344	329	329	329	344	329	344	329
116	370	370	370	370	344	344	329	329	329	344	329	344	329
117	384	384	384	384	359	359	345	345	345	359	345	359	345
118	384	384	384	384	364	359	350	345	345	359	345	359	345
119	340	333	296	185	379	373	367	361	354	347	340	333	326
120	416	416	416	416	391	391	378	378	378	391	378	391	378
121	416	416	416	416	391	391	378	378	378	391	378	391	378
122	416	416	416	416	391	391	378	378	378	391	378	391	378
123	416	416	416	416	391	391	378	378	378	391	378	391	378
124	416	416	416	416	391	391	378	378	378	391	378	391	378
125	416	416	416	416	391	391	378	378	378	391	378	391	378
126	416	416	416	416	391	391	378	378	378	391	378	391	378
127	416	416	416	416	391	391	378	378	378	391	378	391	378
128	416	416	416	416	391	391	378	378	378	391	378	391	378
129	416	416	416	416	391	391	378	378	378	391	378	391	378
130	416	416	416	416	391	391	378	378	378	391	378	391	378
131	416	416	416	416	391	391	378	378	378	391	378	391	378
132	416	416	416	416	391	391	378	378	378	391	378	391	378
133	416	416	416	416	391	391	378	378	378	391	378	391	378
134	416	416	416	416	391	391	378	378	378	391	378	391	378
135	416	416	416	416	391	391	378	378	378	391	378	391	378
136	416	416	416	416	391	391	378	378	378	391	378	391	378
137	416	416	416	416	391	391	378	378	378	391	378	391	378
138	416	416	416	416	391	391	378	378	378	391	378	391	378
139	416	416	416	416	391	391	378	378	378	391	378	391	378
140	416	416	416	416	391	391	378	378	378	391	378	391	378

141	416	416	416	416	391	391	378	378	378	391	378	391	378
142	416	416	416	416	391	391	378	378	378	391	378	391	378
143	416	416	416	416	391	391	378	378	378	391	378	391	378
144	416	416	416	416	391	391	378	378	378	391	378	391	378
145	416	416	416	416	391	391	378	378	378	391	378	391	378
146	416	416	416	416	391	391	378	378	378	391	378	391	378
147	345	345	345	345	316	316	300	300	300	316	300	316	300
148	345	345	345	345	316	316	300	300	300	316	300	316	300
149	368	368	368	368	382	376	370	364	357	350	343	341	329
150	368	368	368	368	372	366	359	352	345	341	331	341	326
151	370	370	370	370	362	355	348	341	334	344	329	344	329
152	370	370	370	370	344	344	329	329	329	344	329	344	329
153	370	370	370	370	344	344	329	329	329	344	329	344	329
154	370	370	370	370	344	344	329	329	329	344	329	344	329
155	370	370	370	370	344	344	329	329	329	344	329	344	329
156	370	370	370	370	344	344	329	329	329	344	329	344	329
157	370	370	370	370	344	344	329	329	329	344	329	344	329
158	370	370	370	370	344	344	329	329	329	344	329	344	329
159	370	370	370	370	344	344	329	329	329	344	329	344	329
160	370	370	370	370	344	344	329	329	329	344	329	344	329
161	370	370	370	370	344	344	329	329	329	344	329	344	329
162	384	384	384	384	359	359	345	345	345	359	345	359	345
163	384	384	384	384	359	359	345	345	345	359	345	359	345
164	384	384	384	384	359	359	345	345	345	359	345	359	345
165	384	384	384	384	359	359	345	345	345	359	345	359	345
166	384	384	384	384	359	359	345	345	345	359	345	359	345
167	384	384	384	384	359	359	345	345	345	359	345	359	345
168	384	384	384	384	359	359	345	345	345	359	345	359	345
169	384	384	384	384	359	359	345	345	345	359	345	359	345
170	384	384	384	384	359	359	345	345	345	359	345	359	345
171	384	384	384	384	359	359	345	345	345	359	345	359	345
172	384	384	384	384	359	359	345	345	345	359	345	359	345
173	384	384	384	384	359	359	345	345	345	359	345	359	345
174	384	384	384	384	359	359	345	345	345	359	345	359	345
175	384	384	384	384	359	359	345	345	345	359	345	359	345

VECTOR

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
DISTANCIAS													
113	**177**	**248**	**342**	**113**	**127**	**113**	**113**	**127**	**127**	**127**	**127**	**127**	**127**

SOLUCIÓN

26361

14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
162	187	177	189	202	213	234	245	256	266	274	282	289	296	311
162	187	167	187	189	202	224	234	245	256	266	274	282	289	304
162	187	162	187	162	187	167	177	189	202	213	224	234	245	266
277	269	259	248	238	227	205	194	187	170	187	162	187	162	187
223	234	245	255	266	274	289	296	304	311	319	326	333	340	354
213	223	234	245	255	266	281	289	296	304	311	319	326	333	347
201	213	223	234	245	255	274	281	289	296	304	311	319	326	340
189	202	213	224	234	245	266	274	282	289	296	304	311	319	333
177	189	202	213	224	234	256	266	274	282	289	296	304	311	326
167	177	189	202	213	224	245	256	266	274	282	289	296	304	319
149	167	177	189	202	213	234	245	256	266	274	282	289	296	311
134	149	167	177	189	202	224	234	245	256	266	274	282	289	304
127	134	149	167	177	189	213	224	234	245	256	266	274	282	296
0	127	134	149	167	177	202	213	224	234	245	256	266	274	289
127	0	127	134	149	167	189	202	213	224	234	245	256	266	282
134	127	0	127	134	149	177	189	202	213	224	234	245	256	274
149	134	127	0	127	134	167	177	189	202	213	224	234	245	266
167	149	134	127	0	127	149	167	177	189	202	213	224	234	256
177	167	149	134	127	0	134	149	167	177	189	202	213	224	245
202	189	177	167	149	134	0	113	134	149	167	177	189	202	224
213	202	189	177	167	149	113	0	127	134	149	167	177	189	213
224	213	202	189	177	167	134	127	0	127	134	149	167	177	202
234	224	213	202	189	177	149	134	127	0	127	134	149	167	189
245	234	224	213	202	189	167	149	134	127	0	127	134	149	177
256	245	234	224	213	202	177	167	149	134	127	0	127	134	167
266	256	245	234	224	213	189	177	167	149	134	127	0	127	149
274	266	256	245	234	224	202	189	177	167	149	134	127	0	134
289	282	274	266	256	245	224	213	202	189	177	167	149	134	0
296	289	282	274	266	256	234	224	213	202	189	177	167	149	127
304	296	289	282	274	266	245	234	224	213	202	189	177	167	134
311	304	296	289	282	274	256	245	234	224	213	202	189	177	149
319	311	304	296	289	282	266	256	245	234	224	213	202	189	167
223	234	245	255	266	274	289	296	304	311	319	326	333	340	354
213	223	234	245	255	266	281	289	296	304	311	319	326	333	347
220	213	223	234	245	255	274	281	289	296	304	311	319	326	340
202	209	220	231	241	252	272	279	286	294	301	309	316	323	337
220	202	220	224	234	245	266	274	282	289	296	304	311	319	333
280	263	280	263	280	263	280	280	276	283	290	298	305	313	327
220	196	220	213	224	234	256	266	274	282	289	296	304	311	326
220	196	220	202	220	224	245	256	266	274	282	289	296	304	319
220	196	220	198	220	221	242	253	263	272	280	287	294	302	317
241	218	241	218	241	218	241	245	256	266	274	282	289	296	311
241	218	241	218	241	218	241	241	248	258	269	276	283	291	306
241	218	241	218	241	218	241	241	245	256	266	274	282	289	304

241	218	241	218	241	218	241	241	234	245	256	266	274	282	296
241	218	241	218	241	218	241	241	232	243	253	264	273	280	295
241	218	241	218	241	218	241	241	224	241	245	256	266	274	289
241	218	241	218	241	218	241	241	218	241	238	248	259	269	284
220	196	220	196	220	196	220	220	213	224	234	245	256	266	282
202	177	202	177	202	177	202	202	202	213	224	234	245	256	274
202	177	202	177	202	177	202	202	200	211	222	233	243	254	273
220	196	220	196	220	196	220	220	196	220	213	224	234	245	266
202	177	202	177	202	177	202	202	181	202	206	217	228	238	260
220	196	220	196	220	196	220	220	196	220	202	220	224	234	256
202	177	202	177	202	177	202	202	177	202	189	202	213	224	245
220	196	220	196	220	196	220	220	196	220	196	220	202	220	234
202	181	202	177	202	177	202	202	177	202	177	202	198	209	231
220	196	220	196	220	196	220	220	196	220	196	220	196	220	224
211	199	202	177	202	177	202	202	177	202	177	202	179	202	215
220	202	220	196	220	196	220	220	196	220	196	220	196	220	213
224	213	202	189	202	177	202	202	177	202	177	202	177	202	202
226	216	220	196	220	196	220	220	196	220	196	220	196	220	198
234	224	213	202	189	177	149	134	113	127	113	134	149	167	189
242	231	221	210	202	186	202	202	177	202	177	202	177	202	180
245	234	224	213	202	189	167	149	134	127	70	127	134	149	177
256	245	234	224	220	202	220	220	196	220	196	220	196	220	196
258	247	236	226	215	204	202	202	177	202	177	202	177	202	177
266	256	245	234	224	213	220	220	196	220	196	220	196	220	196
272	263	252	241	241	220	241	241	218	241	218	241	218	241	218
274	266	256	245	241	224	241	241	218	241	218	241	218	241	218
282	274	266	256	245	234	241	241	218	241	218	241	218	241	218
283	275	268	257	246	236	241	241	218	241	218	241	218	241	218
289	282	274	266	256	245	241	241	218	241	218	241	218	241	218
293	286	279	271	262	251	241	241	218	241	218	241	218	241	218
296	289	282	274	266	256	241	241	218	241	218	241	218	241	218
304	296	289	282	274	266	245	241	224	241	218	241	218	241	218
311	304	296	289	282	274	256	245	234	241	218	241	218	241	218
283	266	283	266	283	280	295	303	310	318	325	332	339	346	360
262	240	262	240	262	261	278	285	293	300	308	315	322	329	343
289	272	289	272	289	272	289	289	282	289	296	304	311	319	333
289	272	289	272	289	272	289	289	272	289	285	293	300	308	322
241	218	241	218	241	218	241	245	256	266	274	282	289	296	311
234	211	234	211	234	211	234	234	240	250	261	270	278	285	300
253	230	253	230	253	230	253	253	230	253	245	256	266	274	289
289	272	289	272	289	272	289	289	272	289	272	289	272	289	278
253	230	253	230	253	230	253	253	230	253	230	253	234	253	266
234	211	234	211	234	211	234	234	211	234	211	234	218	234	250
253	230	253	230	253	230	253	253	230	253	230	253	230	253	230
289	272	289	272	289	272	289	289	272	289	272	289	272	289	272
283	266	283	266	283	266	283	283	266	283	266	283	266	283	266
262	240	262	240	262	240	262	262	240	262	240	262	240	262	240
280	263	280	263	280	263	280	280	263	280	263	280	263	280	263

283	269	283	266	283	266	283	283	266	283	266	283	266	283	266
288	280	283	266	283	266	283	283	266	283	266	283	266	283	266
299	291	284	277	283	266	283	283	266	283	266	283	266	283	266
316	300	316	300	316	300	316	316	310	318	325	332	339	346	360
262	240	262	240	262	240	262	262	270	278	285	293	300	308	322
283	266	283	266	283	266	283	283	266	283	274	283	289	296	311
283	266	283	266	283	266	283	283	266	283	266	283	266	283	289
262	240	262	240	262	240	262	262	240	262	240	262	250	262	278
280	263	280	263	280	263	280	280	263	280	263	280	263	280	266
262	240	262	240	262	240	262	262	240	262	240	262	240	262	250
283	266	283	266	283	266	283	283	266	283	266	283	266	283	266
283	266	283	266	283	266	283	283	266	283	266	283	266	283	266
280	263	280	263	280	263	280	280	263	280	263	280	263	280	263
264	254	262	240	262	240	262	262	240	262	240	262	240	262	240
283	269	283	266	283	266	283	283	266	283	266	283	266	283	266
288	280	283	266	283	266	283	283	266	283	266	283	266	283	266
299	291	284	277	283	266	283	283	266	283	266	283	266	283	266
316	300	316	300	316	300	316	316	300	316	300	316	300	316	300
316	300	316	300	316	300	316	316	300	316	300	316	300	316	300
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
319	311	304	296	289	282	266	256	245	234	224	213	202	189	167
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378

391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
391	378	391	378	391	378	391	391	378	391	378	391	378	391	378
316	300	316	300	316	300	316	316	300	316	300	316	300	316	300
316	300	316	300	316	300	316	316	300	316	300	316	300	316	300
341	326	341	326	341	326	341	341	326	341	326	341	326	341	326
341	326	341	326	341	326	341	341	326	341	326	341	326	341	326
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
344	329	344	329	344	329	344	344	329	344	329	344	329	344	329
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345
359	345	359	345	359	345	359	359	345	359	345	359	345	359	345

15	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**
127	**127**	**127**	**127**	**134**	**113**	**127**	**127**	**127**	**127**	**127**	**127**	**134**	**127**	**127**

29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
319	326	333	340	262	246	262	246	262	312	262	262	262	278	278
311	319	326	333	262	246	262	246	262	312	262	262	262	278	278
274	282	289	296	262	246	262	246	262	312	262	262	262	278	278
162	187	187	185	342	335	328	326	321	315	314	306	304	299	293
361	367	373	379	220	202	220	202	220	280	220	220	220	241	241
354	361	367	373	220	202	220	202	220	280	220	220	220	241	241
347	354	361	367	196	177	196	177	196	263	196	196	196	218	218
340	347	354	361	196	177	196	177	196	263	196	196	196	218	218
333	340	347	354	196	177	196	177	196	263	196	196	196	218	218
326	333	340	347	220	202	220	202	220	280	220	220	220	241	241
319	326	333	340	196	177	196	177	196	263	196	196	196	218	218
311	319	326	333	220	202	220	202	220	280	220	220	220	241	241
304	311	319	326	213	201	196	185	196	263	196	196	196	218	218
296	304	311	319	223	213	220	202	220	280	220	220	220	241	241
289	296	304	311	234	223	213	209	202	263	196	196	196	218	218
282	289	296	304	245	234	223	220	220	280	220	220	220	241	241
274	282	289	296	255	245	234	231	224	263	213	202	198	218	218
266	274	282	289	266	255	245	241	234	280	224	220	220	241	241
256	266	274	282	274	266	255	252	245	263	234	224	221	218	218
234	245	256	266	289	281	274	272	266	280	256	245	242	241	241
224	234	245	256	296	289	281	279	274	280	266	256	253	245	241
213	224	234	245	304	296	289	286	282	276	274	266	263	256	248
202	213	224	234	311	304	296	294	289	283	282	274	272	266	258
189	202	213	224	319	311	304	301	296	290	289	282	280	274	269
177	189	202	213	326	319	311	309	304	298	296	289	287	282	276
167	177	189	202	333	326	319	316	311	305	304	296	294	289	283
149	167	177	189	340	333	326	323	319	313	311	304	302	296	291
127	134	149	167	354	347	340	337	333	327	326	319	317	311	306
0	127	134	149	361	354	347	344	340	334	333	326	324	319	313
127	0	113	134	367	361	354	351	347	341	340	333	331	326	321
134	113	0	127	373	367	361	358	354	348	347	340	338	333	328
149	134	127	0	379	373	367	365	361	355	354	347	345	340	335
361	367	373	379	0	121	134	139	149	189	167	177	180	189	198
354	361	367	373	121	0	121	126	134	208	149	167	170	177	186
347	354	361	367	134	121	0	121	113	189	134	149	154	167	174
344	351	358	365	139	126	121	0	121	208	130	143	148	162	171
340	347	354	361	149	134	113	121	0	189	113	134	138	149	163
334	341	348	355	189	208	189	208	189	0	189	189	189	166	166
333	340	347	354	167	149	134	130	113	189	0	113	124	134	145
326	333	340	347	177	167	149	143	134	189	113	0	81	125	131
324	331	338	345	180	170	154	148	138	189	124	81	0	125	128
319	326	333	340	189	177	167	162	149	166	134	125	125	0	102
313	321	328	335	198	186	174	171	163	166	145	131	128	102	0
311	319	326	333	201	189	177	174	167	166	149	134	131	113	81

304	311	319	326	213	201	189	185	177	169	167	149	144	134	124
302	310	317	324	215	204	192	187	179	171	169	153	148	137	127
296	304	311	319	223	213	201	198	189	179	177	167	163	149	138
291	299	306	314	231	220	209	205	198	187	185	174	171	162	148
289	296	304	311	234	223	213	209	202	192	189	177	174	167	154
282	289	296	304	245	234	223	220	213	208	202	189	186	177	170
281	288	295	303	246	236	225	221	214	208	203	191	187	178	171
274	282	289	296	255	245	234	231	224	215	213	202	198	189	180
270	277	284	292	262	251	241	237	230	221	219	208	205	197	187
266	274	282	289	266	255	245	241	234	226	224	213	210	202	193
256	266	274	282	274	266	255	252	245	236	234	224	221	213	205
245	256	266	274	281	274	266	263	256	247	245	234	231	224	216
241	252	263	272	284	277	269	266	259	251	248	238	235	227	219
234	245	256	266	289	281	274	272	266	258	256	245	242	234	226
226	236	247	258	295	287	280	278	273	266	264	253	251	243	235
224	234	245	256	296	289	281	279	274	268	266	256	253	245	237
213	224	234	245	304	296	289	286	282	276	274	266	263	256	248
220	221	231	242	306	298	291	288	284	278	276	269	266	258	251
202	213	224	234	311	304	296	294	289	283	282	274	272	266	258
202	205	216	226	317	309	302	299	294	288	287	280	278	272	266
189	202	213	224	319	311	304	301	296	290	289	282	280	274	269
220	196	202	220	326	319	311	309	304	298	296	289	287	282	276
202	187	199	211	327	320	313	310	305	299	298	290	288	283	278
220	196	196	220	333	326	319	316	311	305	304	296	294	289	283
241	218	218	241	337	330	323	321	316	310	309	301	299	294	288
241	218	218	241	340	333	326	323	319	313	311	304	302	296	291
241	218	218	241	347	340	333	330	326	320	319	311	309	304	298
241	218	218	241	348	341	334	331	327	321	320	312	310	305	299
241	218	218	241	354	347	340	337	333	327	326	319	317	311	306
241	218	218	241	358	351	344	342	337	331	330	323	321	316	310
241	218	218	241	361	354	347	344	340	334	333	326	324	319	313
241	218	218	241	367	361	354	351	347	341	340	333	331	326	321
241	218	218	241	373	367	361	358	354	348	347	340	338	333	328
366	372	378	385	193	211	193	211	193	173	193	193	193	200	208
350	357	364	370	162	181	162	181	162	138	162	162	163	172	179
340	347	354	361	202	220	202	220	202	112	202	202	202	179	179
329	336	343	350	202	220	202	220	202	112	202	202	202	179	179
319	326	333	340	189	177	167	162	149	166	134	125	125	70	102
308	315	322	329	207	196	183	179	172	174	159	141	137	128	103
296	304	311	319	223	213	201	198	189	179	177	167	163	149	138
289	293	300	308	240	229	218	220	207	198	202	202	202	179	179
274	282	289	296	255	245	234	231	224	215	213	202	198	189	180
261	270	278	285	270	261	250	247	240	231	229	218	215	207	199
253	247	257	268	288	280	273	270	264	256	254	243	240	232	225
289	272	272	289	299	291	284	282	277	271	269	259	256	248	241
283	266	266	283	310	303	295	293	288	282	280	273	271	264	257
262	240	240	262	321	314	306	304	299	293	291	284	282	277	271
280	263	263	280	332	325	318	315	310	304	303	295	293	288	282

283	266	266	283	342	335	328	326	321	315	314	306	304	299	293
283	266	266	283	353	346	339	336	332	326	325	318	316	310	305
283	266	266	283	363	356	349	347	342	336	335	328	326	321	316
366	372	378	385	241	258	241	258	241	173	241	241	241	220	220
329	336	343	350	172	181	162	181	162	128	162	162	162	128	137
319	326	333	340	193	211	193	211	193	137	193	193	193	170	170
296	304	311	319	223	213	201	211	193	179	193	193	193	170	170
285	293	300	308	240	229	218	215	207	198	196	183	179	172	163
280	282	289	296	255	245	234	231	224	215	213	202	198	189	180
262	270	278	285	270	261	250	247	240	231	229	218	215	207	199
283	266	266	283	288	280	273	270	264	256	254	243	240	232	225
283	266	266	283	299	291	284	282	277	271	269	259	256	248	241
280	263	263	280	321	314	306	304	299	293	291	284	282	277	271
262	240	240	262	332	325	318	315	310	304	303	295	293	288	282
283	266	266	283	342	335	328	326	321	315	314	306	304	299	293
283	266	266	283	353	346	339	336	332	326	325	318	316	310	305
283	266	266	283	363	356	349	347	342	336	335	328	326	321	316
316	300	300	316	334	327	320	318	313	307	305	298	296	290	285
316	300	300	316	364	357	350	348	343	338	336	329	327	322	317
344	330	337	344	276	289	276	289	276	215	276	276	276	261	261
344	329	329	344	276	289	276	289	276	215	276	276	276	261	261
344	329	329	344	276	289	276	289	276	215	276	276	276	261	261
344	329	329	344	276	289	276	289	276	235	276	276	276	261	261
344	329	329	344	291	289	277	289	276	261	276	276	276	261	261
359	345	345	359	315	308	300	307	294	287	294	294	294	279	279
359	345	345	359	364	357	350	348	343	338	336	329	327	322	317
149	134	113	111	379	373	367	365	361	355	354	347	345	340	335
391	378	378	391	386	379	373	371	367	362	361	354	352	347	342
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	373	367	361	358	354	348	347	340	338	333	328
391	378	378	391	367	361	354	351	347	341	340	333	333	326	321
391	378	378	391	361	354	347	345	340	334	333	333	333	320	320
391	378	378	391	354	347	340	345	333	327	333	333	333	320	320
391	378	378	391	347	345	333	345	333	320	333	333	333	320	320
391	378	378	391	340	345	333	345	333	313	333	333	333	320	320
391	378	378	391	333	345	333	345	333	305	333	333	333	320	320
391	378	378	391	333	345	333	345	333	290	333	333	333	320	320
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	379	373	367	365	361	355	354	347	345	340	335
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320

391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
391	378	378	391	333	345	333	345	333	286	333	333	333	320	320
316	300	300	316	241	258	241	258	241	200	241	241	241	220	220
316	300	300	316	252	258	241	258	241	211	241	241	241	220	220
341	326	326	341	382	376	370	368	364	359	357	350	348	343	338
341	326	326	341	372	366	359	357	352	347	345	338	337	331	326
344	329	329	344	362	355	348	345	341	335	334	327	325	320	314
344	329	329	344	341	334	327	324	320	314	312	305	303	298	292
344	329	329	344	331	324	317	315	310	304	303	295	293	288	282
344	329	329	344	322	315	307	305	300	294	293	285	283	278	272
344	329	329	344	312	305	298	295	290	284	283	276	276	268	261
344	329	329	344	293	289	278	289	276	263	276	276	276	261	261
344	329	329	344	283	289	276	289	276	249	276	276	276	261	261
344	329	329	344	276	289	276	289	276	224	276	276	276	261	261
344	329	329	344	276	289	276	289	276	215	276	276	276	261	261
344	329	329	344	276	289	276	289	276	215	276	276	276	261	261
344	329	329	344	303	295	288	289	280	274	276	276	276	261	261
359	345	345	359	294	307	294	307	294	240	294	294	294	279	279
359	345	345	359	347	340	333	330	326	320	319	311	309	304	298
359	345	345	359	337	330	323	321	316	310	309	301	299	294	288
359	345	345	359	319	311	304	307	296	290	294	294	294	279	279
359	345	345	359	309	307	294	307	294	280	294	294	294	279	279
359	345	345	359	299	307	294	307	294	270	294	294	294	279	279
359	345	345	359	294	307	294	307	294	240	294	294	294	279	279
359	345	345	359	294	307	294	307	294	240	294	294	294	279	279
359	345	345	359	294	307	294	307	294	240	294	294	294	279	279
359	345	345	359	294	307	294	307	294	240	294	294	294	279	279
359	345	345	359	294	307	294	307	294	240	294	294	294	279	279
359	345	347	359	294	307	294	307	294	240	294	294	294	279	279
359	349	356	363	294	307	294	307	294	240	294	294	294	279	279
359	345	345	359	356	349	342	340	335	329	328	321	319	314	308

30	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**
113	**127**	**379**	**121**	**121**	**121**	**121**	**189**	**189**	**113**	**81**	**125**	**102**	**81**	**113**

44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
278	278	278	278	278	262	246	246	262	246	262	246	262	246	262
278	278	278	278	278	262	246	246	262	246	262	246	262	246	262
278	278	278	278	278	262	246	246	262	246	262	246	262	246	262
291	284	283	278	278	269	259	258	262	246	262	246	262	246	262
241	241	241	241	241	234	245	246	255	262	266	274	281	284	289
241	241	241	241	241	223	234	236	245	251	255	266	274	277	281
218	218	218	218	218	213	223	225	234	241	245	255	266	269	274
218	218	218	218	218	202	213	214	224	230	234	245	256	259	266
218	218	218	218	218	196	202	203	213	219	224	234	245	248	256
241	241	241	241	241	220	202	202	220	208	220	224	234	238	245
218	218	218	218	218	196	177	178	196	197	202	213	224	227	234
241	241	241	241	241	220	202	202	220	202	220	202	220	216	224
218	218	218	218	218	196	177	177	196	177	196	189	202	205	213
241	241	241	241	241	220	202	202	220	202	220	202	220	202	220
218	218	218	218	218	196	177	177	196	177	196	177	196	181	196
241	241	241	241	241	220	202	202	220	202	220	202	220	202	220
218	218	218	218	218	196	177	177	196	177	196	177	196	177	196
241	241	241	241	241	220	202	202	220	202	220	202	220	202	220
218	218	218	218	218	196	177	177	196	177	196	177	196	177	196
241	241	241	241	241	220	202	202	220	202	220	202	220	202	220
241	241	241	241	241	220	202	202	220	202	220	202	220	202	220
245	234	232	224	218	213	202	200	196	181	196	177	196	177	196
256	245	243	241	241	224	213	211	220	202	220	202	220	202	220
266	256	253	245	238	234	224	222	213	206	202	189	196	177	196
274	266	264	256	248	245	234	233	224	217	220	202	220	202	220
282	274	273	266	259	256	245	243	234	228	224	213	202	198	196
289	282	280	274	269	266	256	254	245	238	234	224	220	209	220
304	296	295	289	284	282	274	273	266	260	256	245	234	231	224
311	304	302	296	291	289	282	281	274	270	266	256	245	241	234
319	311	310	304	299	296	289	288	282	277	274	266	256	252	245
326	319	317	311	306	304	296	295	289	284	282	274	266	263	256
333	326	324	319	314	311	304	303	296	292	289	282	274	272	266
201	213	215	223	231	234	245	246	255	262	266	274	281	284	289
189	201	204	213	220	223	234	236	245	251	255	266	274	277	281
177	189	192	201	209	213	223	225	234	241	245	255	266	269	274
174	185	187	198	205	209	220	221	231	237	241	252	263	266	272
167	177	179	189	198	202	213	214	224	230	234	245	256	259	266
166	169	171	179	187	192	208	208	215	221	226	236	247	251	258
149	167	169	177	185	189	202	203	213	219	224	234	245	248	256
134	149	153	167	174	177	189	191	202	208	213	224	234	238	245
131	144	148	163	171	174	186	187	198	205	210	221	231	235	242
113	134	137	149	162	167	177	178	189	197	202	213	224	227	234
81	124	127	138	148	154	170	171	180	187	193	205	216	219	226
0	113	121	134	143	149	167	168	177	184	189	202	213	216	224

113	0	75	113	130	134	154	154	167	173	177	189	202	205	213
121	75	0	104	128	132	154	154	164	171	175	187	199	203	211
134	113	104	0	99	125	154	154	149	160	167	177	189	194	202
143	130	128	99	0	125	154	154	139	154	155	170	181	185	194
149	134	132	125	125	0	121	121	134	142	149	167	177	181	189
167	154	154	154	154	121	0	70	121	129	134	149	167	170	177
168	154	154	154	154	121	70	0	121	128	133	147	165	169	176
177	167	164	149	139	134	121	121	0	121	113	134	149	155	167
184	173	171	160	154	142	129	128	121	0	121	127	140	144	157
189	177	175	167	155	149	134	133	113	121	0	121	134	139	149
202	189	187	177	170	167	149	147	134	127	121	0	121	126	134
213	202	199	189	181	177	167	165	149	140	134	121	0	121	113
216	205	203	194	185	181	170	169	155	144	139	126	121	0	121
224	213	211	202	194	189	177	176	167	157	149	134	113	121	0
232	221	219	211	203	199	187	185	175	169	164	146	132	128	121
234	224	221	213	205	202	189	187	177	171	167	149	134	130	113
245	234	232	224	216	213	202	200	189	181	177	167	149	143	134
248	237	235	226	219	216	205	203	193	185	180	170	154	148	138
256	245	243	234	227	224	213	211	202	194	196	177	196	177	196
263	253	251	242	235	231	221	219	210	203	198	186	174	171	163
266	256	253	245	238	234	224	222	213	206	202	189	196	177	196
274	266	264	256	248	245	234	233	224	217	213	202	189	185	177
276	268	266	258	251	247	236	235	226	219	215	204	192	187	179
282	274	273	266	259	256	245	243	234	228	224	213	202	198	189
286	279	278	272	266	263	252	251	241	235	231	220	209	205	198
289	282	280	274	269	266	256	254	245	238	234	224	213	209	202
296	289	287	282	277	274	266	265	256	249	245	234	224	220	213
297	290	288	283	278	275	268	266	257	251	246	236	225	221	214
304	296	295	289	284	282	274	273	266	260	256	245	234	231	224
308	301	299	293	288	286	279	278	271	266	262	251	241	237	230
311	304	302	296	291	289	282	281	274	270	266	256	245	241	234
319	311	310	304	299	296	289	288	282	277	274	266	256	252	245
326	319	317	311	306	304	296	295	289	284	282	274	266	263	256
211	222	224	232	239	243	254	255	264	270	273	280	288	290	295
183	196	198	207	215	218	229	230	239	246	250	261	270	273	278
179	179	179	189	198	202	220	220	224	230	234	245	256	259	266
179	179	179	179	179	202	220	220	207	220	218	229	240	243	250
113	134	137	149	162	167	177	178	189	197	202	213	224	227	234
92	92	100	128	137	141	158	161	172	178	183	196	207	211	218
134	113	112	112	112	143	170	170	149	170	167	177	189	194	202
179	179	179	179	179	202	220	220	202	220	202	220	202	220	202
177	167	164	149	139	143	170	170	143	170	143	170	149	170	167
196	183	180	172	165	159	141	140	128	140	118	140	128	140	141
222	211	209	200	191	187	175	174	165	170	146	170	143	170	143
238	227	225	216	209	205	220	220	202	220	202	220	202	220	202
254	243	241	232	225	222	211	211	200	211	193	211	193	211	193
269	259	257	248	241	238	227	226	216	210	205	194	181	181	170
280	273	271	264	257	254	243	242	232	226	222	211	200	208	189

291	284	283	277	272	269	259	258	248	242	238	227	216	213	205
303	295	294	288	283	280	273	272	264	258	254	243	232	229	222
314	306	305	299	294	291	284	283	277	272	269	259	248	245	238
220	222	224	232	239	243	258	258	264	270	273	280	288	290	295
141	158	162	172	179	183	196	197	207	214	218	229	240	243	250
170	170	170	170	170	193	211	211	193	211	202	213	224	227	234
170	170	170	170	170	193	211	211	193	211	193	211	193	211	202
159	141	138	128	127	162	181	181	162	181	162	181	172	181	183
177	167	166	166	166	189	208	208	189	208	189	208	189	208	189
196	183	180	172	165	162	181	181	162	181	162	181	162	181	162
222	211	209	200	191	193	211	211	193	211	193	211	193	211	193
238	227	225	216	209	205	211	211	193	211	193	211	193	211	193
269	259	257	248	241	238	227	226	216	210	205	208	189	208	189
280	273	271	264	257	254	243	242	232	226	222	211	200	196	187
291	284	283	277	272	269	259	258	248	242	238	227	216	213	205
303	295	294	288	283	280	273	272	264	258	254	243	232	229	222
314	306	305	299	294	291	284	283	277	272	269	259	248	245	238
283	275	274	268	261	257	258	258	241	258	241	258	241	258	241
315	308	306	300	295	293	285	284	278	273	270	261	250	258	241
261	261	261	261	261	276	289	289	276	289	276	289	276	289	276
261	261	261	261	261	276	289	289	276	289	276	289	276	289	276
261	261	261	261	261	276	289	289	276	289	276	289	276	289	276
261	261	261	261	261	276	289	289	276	289	276	289	276	289	276
261	261	261	261	261	276	289	289	276	289	276	289	276	289	276
279	279	279	279	279	294	307	307	294	307	294	307	294	307	294
315	308	306	300	295	294	307	307	294	307	294	307	294	307	294
333	326	324	319	314	311	304	303	296	292	289	282	274	272	266
340	333	331	326	321	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
326	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
333	326	324	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333

320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
320	320	320	320	320	333	345	345	333	345	333	345	333	345	333
220	220	220	220	220	241	258	258	241	258	241	258	241	258	241
220	220	220	220	220	241	258	258	241	258	241	258	241	258	241
336	329	328	322	317	315	308	307	300	296	293	287	278	287	274
324	317	316	310	305	303	295	294	288	287	280	287	274	287	274
312	305	304	298	293	290	289	289	276	289	276	289	276	289	276
290	283	281	275	270	276	289	289	276	289	276	289	276	289	276
280	273	271	264	261	276	289	289	276	289	276	289	276	289	276
270	261	261	261	261	276	289	289	276	289	276	289	276	289	276
261	261	261	261	261	276	289	289	276	289	276	289	276	289	276
261	261	261	261	261	276	289	289	276	289	276	289	276	289	276
261	261	261	261	261	276	289	289	276	289	276	289	276	289	276
261	261	261	261	261	276	289	289	276	289	276	289	276	289	276
261	261	261	261	261	276	289	289	276	289	276	289	276	289	276
261	261	261	261	261	276	289	289	276	289	276	289	276	289	276
261	261	261	261	261	276	289	289	276	289	276	289	276	289	276
279	279	279	279	279	294	307	307	294	307	294	307	294	307	294
296	289	287	281	279	294	307	307	294	307	294	307	294	307	294
286	279	279	279	279	294	307	307	294	307	294	307	294	307	294
279	279	279	279	279	294	307	307	294	307	294	307	294	307	294
279	279	279	279	279	294	307	307	294	307	294	307	294	307	294
279	279	279	279	279	294	307	307	294	307	294	307	294	307	294
279	279	279	279	279	294	307	307	294	307	294	307	294	307	294
279	279	279	279	279	294	307	307	294	307	294	307	294	307	294
279	279	279	279	279	294	307	307	294	307	294	307	294	307	294
279	279	279	279	279	294	307	307	294	307	294	307	294	307	294
279	279	279	279	279	294	307	307	294	307	294	307	294	307	294
279	279	279	279	279	294	307	307	294	307	294	307	294	307	294
279	279	279	279	279	294	307	307	294	307	294	307	294	307	294
306	299	297	291	286	294	307	307	294	307	294	307	294	307	294

45	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**
75	**104**	**99**	**125**	**121**	**70**	**121**	**121**	**121**	**121**	**121**	**121**	**121**	**121**	**121**

59	60	61	62	63	64	65	66	67	68	69	70	71	72	73
246	262	256	262	266	272	274	282	283	289	294	296	304	305	311
246	262	246	262	256	263	266	274	276	282	286	289	296	297	304
246	262	246	262	202	246	213	262	246	262	278	278	278	278	278
246	262	246	262	187	246	187	262	246	262	278	278	278	278	278
295	296	304	306	311	317	319	326	327	333	337	340	347	348	354
287	289	296	298	304	309	311	319	320	326	330	333	340	341	347
280	281	289	291	296	302	304	311	313	319	323	326	333	334	340
273	274	282	284	289	294	296	304	305	311	316	319	326	327	333
264	266	274	276	282	287	289	296	298	304	309	311	319	320	326
253	256	266	269	274	280	282	289	290	296	301	304	311	312	319
243	245	256	258	266	272	274	282	283	289	294	296	304	305	311
232	234	245	248	256	263	266	274	276	282	286	289	296	297	304
221	224	234	237	245	253	256	266	268	274	279	282	289	290	296
211	220	224	226	234	242	245	256	258	266	272	274	282	283	289
199	202	213	216	224	231	234	245	247	256	263	266	274	275	282
202	220	202	220	213	221	224	234	236	245	252	256	266	268	274
177	196	189	196	202	210	213	224	226	234	241	245	256	257	266
202	220	202	220	189	202	202	220	215	224	241	241	245	246	256
177	196	177	196	177	186	189	202	204	213	220	224	234	236	245
202	220	202	220	149	202	167	220	202	220	241	241	241	241	241
202	220	202	220	134	202	149	220	202	220	241	241	241	241	241
177	196	177	196	113	177	134	196	177	196	218	218	218	218	218
202	220	202	220	127	202	127	220	202	220	241	241	241	241	241
177	196	177	196	113	177	70	196	177	196	218	218	218	218	218
202	220	202	220	134	202	127	220	202	220	241	241	241	241	241
179	196	177	196	149	177	134	196	177	196	218	218	218	218	218
202	220	202	220	167	202	149	220	202	220	241	241	241	241	241
215	213	202	198	189	180	177	196	177	196	218	218	218	218	218
226	224	213	220	202	202	189	220	202	220	241	241	241	241	241
236	234	224	221	213	205	202	196	187	196	218	218	218	218	218
247	245	234	231	224	216	213	202	199	196	218	218	218	218	218
258	256	245	242	234	226	224	220	211	220	241	241	241	241	241
295	296	304	306	311	317	319	326	327	333	337	340	347	348	354
287	289	296	298	304	309	311	319	320	326	330	333	340	341	347
280	281	289	291	296	302	304	311	313	319	323	326	333	334	340
278	279	286	288	294	299	301	309	310	316	321	323	330	331	337
273	274	282	284	289	294	296	304	305	311	316	319	326	327	333
266	268	276	278	283	288	290	298	299	305	310	313	320	321	327
264	266	274	276	282	287	289	296	298	304	309	311	319	320	326
253	256	266	269	274	280	282	289	290	296	301	304	311	312	319
251	253	263	266	272	278	280	287	288	294	299	302	309	310	317
243	245	256	258	266	272	274	282	283	289	294	296	304	305	311
235	237	248	251	258	266	269	276	278	283	288	291	298	299	306
232	234	245	248	256	263	266	274	276	282	286	289	296	297	304

221	224	234	237	245	253	256	266	268	274	279	282	289	290	296
219	221	232	235	243	251	253	264	266	273	278	280	287	288	295
211	213	224	226	234	242	245	256	258	266	272	274	282	283	289
203	205	216	219	227	235	238	248	251	259	266	269	277	278	284
199	202	213	216	224	231	234	245	247	256	263	266	274	275	282
187	189	202	205	213	221	224	234	236	245	252	256	266	268	274
185	187	200	203	211	219	222	233	235	243	251	254	265	266	273
175	177	189	193	202	210	213	224	226	234	241	245	256	257	266
169	171	181	185	194	203	206	217	219	228	235	238	249	251	260
164	167	177	180	196	198	202	213	215	224	231	234	245	246	256
146	149	167	170	177	186	189	202	204	213	220	224	234	236	245
132	134	149	154	196	174	196	189	192	202	209	213	224	225	234
128	130	143	148	177	171	177	185	187	198	205	209	220	221	231
121	113	134	138	196	163	196	177	179	189	198	202	213	214	224
0	121	121	128	177	148	177	169	171	179	187	192	204	205	215
121	0	121	124	196	145	196	167	169	177	185	189	202	203	213
121	121	0	121	177	131	177	149	153	167	174	177	189	191	202
128	124	121	0	196	128	196	144	148	163	171	174	186	187	198
177	196	177	196	0	177	113	196	177	196	218	218	218	218	218
148	145	131	128	177	0	177	124	127	138	154	154	170	171	180
177	196	177	196	113	177	0	196	177	196	218	218	218	218	218
169	167	149	144	196	124	196	0	121	113	130	134	149	152	167
171	169	153	148	177	127	177	121	0	121	154	154	154	154	164
179	177	167	163	196	138	196	113	121	0	125	125	134	136	149
187	185	174	171	218	154	218	130	154	125	0	84	126	128	139
192	189	177	174	218	154	218	134	154	125	84	0	113	118	134
204	202	189	186	218	170	218	149	154	134	126	113	0	70	113
205	203	191	187	218	171	218	152	154	136	128	118	70	0	107
215	213	202	198	218	180	218	167	164	149	139	134	113	107	0
221	219	208	205	218	187	218	173	171	160	148	142	129	128	96
226	224	213	210	218	193	218	177	175	167	155	149	134	133	113
236	234	224	221	218	205	218	189	187	177	170	167	149	147	134
247	245	234	231	224	216	218	202	199	189	181	177	167	165	149
301	303	310	312	318	323	325	332	333	339	343	346	353	354	360
284	285	293	295	300	306	308	315	317	322	327	329	336	337	343
273	274	282	284	289	294	296	304	305	311	316	319	326	327	333
259	261	270	272	278	283	285	293	294	300	305	308	315	316	322
243	245	256	258	266	272	274	282	283	289	294	296	304	305	311
227	229	240	242	250	258	261	270	272	278	283	285	293	294	300
211	213	224	226	234	242	245	256	258	266	272	274	282	283	289
220	202	220	210	272	226	272	240	242	250	257	261	270	271	278
175	177	189	193	230	210	230	224	226	234	241	245	256	257	266
155	158	172	175	211	192	211	207	209	218	225	229	240	241	250
170	143	170	143	230	170	230	179	181	191	200	203	215	216	225
220	202	220	202	272	220	272	202	220	202	181	185	198	199	209
211	193	211	193	266	211	266	193	211	193	170	170	179	180	191
181	162	181	162	240	181	240	162	181	162	139	143	162	164	174
208	189	208	189	263	208	263	189	208	189	166	166	166	166	166

211	194	211	193	266	211	266	193	211	193	170	170	170	170	170
213	211	211	197	266	211	266	193	211	193	170	170	170	170	170
229	227	216	214	266	211	266	193	211	193	170	170	170	170	170
301	303	310	312	318	323	325	332	333	339	343	346	353	354	360
259	261	270	272	278	283	285	293	294	300	305	308	315	316	322
243	245	256	258	266	272	274	282	283	289	294	296	304	305	311
211	213	224	226	266	242	266	256	258	266	272	274	282	283	289
193	196	207	210	240	226	240	240	242	250	257	261	270	271	278
208	189	208	193	263	210	263	224	226	234	241	245	256	257	266
181	162	181	175	240	192	240	207	209	218	225	229	240	241	250
211	193	211	193	266	211	266	193	211	193	200	203	215	216	225
211	193	211	193	266	211	266	193	211	193	181	185	198	199	209
208	189	208	189	263	208	263	189	208	189	166	166	166	166	174
181	175	181	162	240	181	240	162	181	162	127	127	137	138	152
211	194	211	193	266	211	266	193	211	193	170	170	170	170	170
213	211	211	197	266	211	266	193	211	193	170	170	170	170	170
229	227	216	214	266	211	266	193	211	193	170	170	170	170	170
258	241	258	241	300	258	300	241	258	241	220	220	220	220	220
258	241	258	241	300	258	300	241	258	241	220	220	220	220	220
289	276	289	276	329	289	329	286	289	294	299	301	309	310	316
289	276	289	276	329	289	329	276	289	276	273	275	283	284	290
289	276	289	276	329	289	329	276	289	276	264	268	275	276	283
289	276	289	276	329	289	329	276	289	276	261	261	261	261	261
289	276	289	276	329	289	329	276	289	276	261	261	261	261	261
307	294	307	294	345	307	345	294	307	294	279	279	279	279	279
307	294	307	294	345	307	345	294	307	294	279	279	279	279	279
258	256	245	242	234	226	224	213	211	208	229	229	229	229	229
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	334	339	341	348	349	355
345	333	345	333	378	345	378	333	345	333	332	334	341	342	348
345	333	345	333	378	345	378	333	345	333	325	327	334	335	341
345	333	345	333	378	345	378	333	345	333	320	320	327	328	334
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320

345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	321	327
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
345	333	345	333	378	345	378	333	345	333	320	320	320	320	320
258	241	258	241	300	258	300	241	258	249	256	259	269	270	277
258	241	258	241	300	258	300	241	258	241	245	249	259	261	269
287	274	287	274	326	287	326	274	287	274	258	258	258	258	258
287	274	287	274	326	287	326	274	287	274	258	258	258	258	258
289	276	289	276	329	289	329	276	289	276	261	261	261	261	261
289	276	289	276	329	289	329	276	289	276	261	261	261	261	261
289	276	289	276	329	289	329	276	289	276	261	261	261	261	261
289	276	289	276	329	289	329	276	289	276	261	261	261	261	261
289	276	289	276	329	289	329	276	289	276	261	261	261	261	261
289	276	289	276	329	289	329	276	289	276	261	261	261	261	261
289	276	289	276	329	289	329	276	289	276	261	261	261	261	261
289	276	289	276	329	289	329	276	289	276	261	261	261	261	261
289	276	289	276	329	289	329	276	289	276	261	265	273	274	280
289	276	289	276	329	289	329	276	289	278	283	285	293	294	300
289	276	289	276	329	289	329	276	289	276	261	261	261	261	261
307	294	307	294	345	307	345	294	307	294	279	279	279	279	279
307	294	307	294	345	307	345	294	307	294	279	279	279	279	279
307	294	307	294	345	307	345	294	307	294	279	279	279	279	279
307	294	307	294	345	307	345	294	307	294	279	279	279	279	279
307	294	307	294	345	307	345	294	307	294	279	279	279	279	279
307	294	307	294	345	307	345	294	307	294	279	279	279	279	279
307	294	307	294	345	307	345	294	307	294	289	292	299	300	307
307	294	307	294	345	307	345	294	307	294	279	279	279	279	279
307	294	307	294	345	307	345	294	307	294	279	279	279	280	287
307	294	307	294	345	307	345	294	307	294	279	282	289	290	297
307	294	307	294	345	307	345	294	307	294	299	302	309	310	316
307	294	307	294	345	307	345	297	307	304	309	312	319	320	326
307	294	307	294	345	307	345	307	308	314	319	321	328	329	335
307	294	307	294	345	307	345	294	307	294	279	279	279	279	279

60	**61**	**62**	**63**	**64**	**65**	**66**	**67**	**68**	**69**	**70**	**71**	**72**	**73**	**74**
121	**121**	**196**	**177**	**177**	**196**	**121**	**121**	**125**	**84**	**113**	**70**	**107**	**96**	**87**

74	75	76	77	78	79	80	81	82	83	84	85	86	87	88
316	319	326	333	314	294	320	320	278	273	287	320	287	273	287
308	311	319	326	314	294	320	320	278	273	287	320	287	273	287
278	278	282	289	314	294	320	320	278	273	287	320	287	273	287
278	278	278	278	348	332	321	320	299	288	287	320	287	273	287
358	361	367	373	283	262	289	289	241	234	253	289	255	270	288
351	354	361	367	283	262	289	289	241	234	253	289	253	261	280
344	347	354	361	266	240	272	272	218	211	230	272	234	250	273
337	340	347	354	266	240	272	272	218	211	230	272	230	240	264
330	333	340	347	266	240	272	272	218	211	230	272	230	229	254
323	326	333	340	283	262	289	289	241	234	253	289	253	234	253
316	319	326	333	266	240	272	272	218	211	230	272	230	211	232
308	311	319	326	283	262	289	289	241	234	253	289	253	234	253
301	304	311	319	266	240	272	272	218	211	230	272	230	211	230
293	296	304	311	283	262	289	289	241	234	253	289	253	234	253
286	289	296	304	266	240	272	272	218	211	230	272	230	211	230
279	282	289	296	283	262	289	289	241	234	253	289	253	234	253
271	274	282	289	266	240	272	272	218	211	230	272	230	211	230
262	266	274	282	283	262	289	289	241	234	253	289	253	234	253
251	256	266	274	280	261	272	272	218	211	230	272	230	211	230
241	241	245	256	295	278	289	289	241	234	253	289	253	234	253
241	241	241	245	303	285	289	289	245	234	253	289	253	234	253
218	218	224	234	310	293	282	272	256	240	230	272	230	211	230
241	241	241	241	318	300	289	289	266	250	253	289	253	234	253
218	218	218	218	325	308	296	285	274	261	245	272	230	211	230
241	241	241	241	332	315	304	293	282	270	256	289	253	234	253
218	218	218	218	339	322	311	300	289	278	266	272	234	218	230
241	241	241	241	346	329	319	308	296	285	274	289	253	234	253
218	218	218	218	360	343	333	322	311	300	289	278	266	250	230
241	241	241	241	366	350	340	329	319	308	296	289	274	261	253
218	218	218	218	372	357	347	336	326	315	304	293	282	270	247
218	218	218	218	378	364	354	343	333	322	311	300	289	278	257
241	241	241	241	385	370	361	350	340	329	319	308	296	285	268
358	361	367	373	193	162	202	202	189	207	223	240	255	270	288
351	354	361	367	211	181	220	220	177	196	213	229	245	261	280
344	347	354	361	193	162	202	202	167	183	201	218	234	250	273
342	344	351	358	211	181	220	220	162	179	198	220	231	247	270
337	340	347	354	193	162	202	202	149	172	189	207	224	240	264
331	334	341	348	173	138	112	112	166	174	179	198	215	231	256
330	333	340	347	193	162	202	202	134	159	177	202	213	229	254
323	326	333	340	193	162	202	202	125	141	167	202	202	218	243
321	324	331	338	193	163	202	202	125	137	163	202	198	215	240
316	319	326	333	200	172	179	179	70	128	149	179	189	207	232
310	313	321	328	208	179	179	179	102	103	138	179	180	199	225
308	311	319	326	211	183	179	179	113	92	134	179	177	196	222

301	304	311	319	222	196	179	179	134	92	113	179	167	183	211
299	302	310	317	224	198	179	179	137	100	112	179	164	180	209
293	296	304	311	232	207	189	179	149	128	112	179	149	172	200
288	291	299	306	239	215	198	179	162	137	112	179	139	165	191
286	289	296	304	243	218	202	202	167	141	143	202	143	159	187
279	282	289	296	254	229	220	220	177	158	170	220	170	141	175
278	281	288	295	255	230	220	220	178	161	170	220	170	140	174
271	274	282	289	264	239	224	207	189	172	149	202	143	128	165
266	270	277	284	270	246	230	220	197	178	170	220	170	140	170
262	266	274	282	273	250	234	218	202	183	167	202	143	118	146
251	256	266	274	280	261	245	229	213	196	177	220	170	140	170
241	245	256	266	288	270	256	240	224	207	189	202	149	128	143
237	241	252	263	290	273	259	243	227	211	194	220	170	140	170
230	234	245	256	295	278	266	250	234	218	202	202	167	141	143
221	226	236	247	301	284	273	259	243	227	211	220	175	155	170
219	224	234	245	303	285	274	261	245	229	213	202	177	158	143
208	213	224	234	310	293	282	270	256	240	224	220	189	172	170
205	210	221	231	312	295	284	272	258	242	226	210	193	175	143
218	218	218	224	318	300	289	278	266	250	234	272	230	211	230
187	193	205	216	323	306	294	283	272	258	242	226	210	192	170
218	218	218	218	325	308	296	285	274	261	245	272	230	211	230
173	177	189	202	332	315	304	293	282	270	256	240	224	207	179
171	175	187	199	333	317	305	294	283	272	258	242	226	209	181
160	167	177	189	339	322	311	300	289	278	266	250	234	218	191
148	155	170	181	343	327	316	305	294	283	272	257	241	225	200
142	149	167	177	346	329	319	308	296	285	274	261	245	229	203
129	134	149	167	353	336	326	315	304	293	282	270	256	240	215
128	133	147	165	354	337	327	316	305	294	283	271	257	241	216
96	113	134	149	360	343	333	322	311	300	289	278	266	250	225
0	87	127	140	364	347	337	326	316	305	293	282	271	257	232
87	0	113	134	366	350	340	329	319	308	296	285	274	261	236
127	113	0	113	372	357	347	336	326	315	304	293	282	270	247
140	134	113	0	378	364	354	343	333	322	311	300	289	278	257
364	366	372	378	0	139	165	181	200	216	232	248	264	277	294
347	350	357	364	139	0	150	150	172	189	207	223	239	255	277
337	340	347	354	165	150	0	128	179	187	189	207	224	240	264
326	329	336	343	181	150	128	0	179	187	172	189	207	224	248
316	319	326	333	200	172	179	179	0	128	149	179	189	207	232
305	308	315	322	216	189	187	187	128	0	128	187	172	189	216
293	296	304	311	232	207	189	172	149	128	0	166	149	172	200
282	285	293	300	248	223	207	189	179	187	166	0	166	187	181
271	274	282	289	264	239	224	207	189	172	149	166	0	128	165
257	261	270	278	277	255	240	224	207	189	172	187	128	0	139
232	236	247	257	294	277	264	248	232	216	200	181	165	139	0
216	220	231	241	305	288	277	264	248	232	216	200	181	187	166
199	203	215	225	316	299	288	277	264	248	232	216	200	181	154
180	185	198	209	327	310	299	288	277	264	248	232	216	200	172
166	169	179	191	337	321	310	299	288	277	264	248	232	216	189

170	170	170	174	348	332	321	310	299	288	277	264	248	232	207
170	170	170	170	358	342	332	321	310	299	288	277	264	248	224
170	170	170	170	368	353	342	332	321	310	299	288	277	264	240
364	366	372	378	166	196	165	181	220	227	232	248	264	277	294
326	329	336	343	181	149	150	150	128	149	172	189	207	224	248
316	319	326	333	200	172	149	128	170	177	154	172	189	207	232
293	296	304	311	232	207	189	172	170	177	154	128	154	177	200
282	285	293	300	248	223	207	189	172	149	128	150	128	149	181
271	274	282	289	264	239	224	207	189	174	149	128	147	174	165
257	261	270	278	277	255	240	224	207	189	172	150	128	136	139
232	236	247	257	294	277	264	248	232	216	200	181	165	177	154
216	220	231	241	305	288	277	264	248	232	216	200	181	177	154
180	185	198	209	327	310	299	288	277	264	248	232	216	200	172
163	169	179	191	337	321	310	299	288	277	264	248	232	216	189
170	170	170	174	348	332	321	310	299	288	277	264	248	232	207
170	170	170	170	358	342	332	321	310	299	288	277	264	248	224
170	170	170	170	368	353	342	332	321	310	299	288	277	264	240
220	220	220	220	340	323	313	301	290	279	268	252	236	227	208
220	220	220	220	369	354	343	333	322	311	300	289	278	266	241
321	323	330	337	211	239	202	202	261	266	249	202	249	266	249
295	298	305	313	231	239	202	202	261	266	249	202	249	266	249
287	290	298	305	241	239	202	202	261	266	249	202	249	266	249
261	261	268	275	279	259	243	227	261	266	249	202	249	266	249
261	261	261	261	298	280	269	254	261	266	249	202	249	266	249
279	279	279	279	321	304	293	282	279	284	270	227	270	284	270
279	279	279	279	369	354	343	333	322	311	300	289	278	284	270
229	229	229	229	385	370	361	350	340	329	319	308	296	285	268
320	320	320	320	391	376	367	357	347	336	326	315	312	324	312
320	320	320	320	284	304	277	277	320	324	312	277	312	324	312
320	320	320	320	378	364	354	343	333	324	312	300	312	324	312
320	320	320	320	372	357	347	336	326	324	312	293	312	324	312
320	320	320	320	366	350	340	329	320	324	312	285	312	324	312
320	320	320	320	360	343	333	322	320	324	312	278	312	324	312
320	320	320	320	353	336	326	315	320	324	312	277	312	324	312
320	320	320	320	346	329	319	308	320	324	312	277	312	324	312
320	320	320	320	339	322	311	300	320	324	312	277	312	324	312
320	320	320	320	325	308	296	285	320	324	312	277	312	324	312
320	320	320	320	318	304	289	278	320	324	312	277	312	324	312
320	320	320	320	310	304	282	277	320	324	312	277	312	324	312
320	320	320	320	295	304	277	277	320	324	312	277	312	324	312
320	320	320	320	288	304	277	277	320	324	312	277	312	324	312
320	320	320	320	385	370	361	350	340	329	319	308	312	324	312
320	320	327	334	284	304	277	277	320	324	312	277	312	324	312
359	362	368	374	284	304	277	277	320	324	312	277	312	324	312
352	355	362	368	284	304	277	277	320	324	312	277	312	324	312
345	348	355	362	284	304	277	277	320	324	312	277	312	324	312
338	341	348	355	284	304	277	277	320	324	312	277	312	324	312
320	320	320	320	284	304	277	277	320	324	312	277	312	324	312

324	327	334	341	284	304	277	277	320	324	312	277	312	324	312
331	334	341	348	284	304	277	277	320	324	312	277	312	324	312
320	320	320	327	284	304	277	277	320	324	312	277	312	324	312
320	320	320	320	284	304	277	277	320	324	312	277	312	324	312
320	320	320	320	284	304	277	277	320	324	312	277	312	324	312
320	320	320	320	284	304	277	277	320	324	312	277	312	324	312
281	284	292	299	250	225	209	191	220	227	208	154	208	227	208
274	277	284	292	261	236	220	203	220	227	208	154	208	227	208
258	258	258	258	388	373	364	354	343	333	322	311	300	289	271
258	258	258	258	377	363	352	342	331	321	310	299	288	276	255
261	261	261	261	367	351	341	330	320	309	298	286	275	266	249
261	261	261	261	347	330	320	309	298	286	275	262	249	266	249
261	261	261	261	337	321	310	299	288	276	264	248	249	266	249
261	261	261	261	328	311	300	289	278	266	250	234	249	266	249
261	261	261	261	319	301	290	279	268	266	249	220	249	266	249
261	261	261	261	299	281	270	255	261	266	249	202	249	266	249
261	261	261	265	289	271	257	241	261	266	249	202	249	266	249
264	268	275	283	271	248	232	216	261	266	249	202	249	266	249
285	288	295	303	245	239	203	202	261	266	249	202	249	266	249
305	308	315	322	216	239	202	202	261	266	249	202	249	266	249
261	261	261	261	309	291	280	269	261	266	249	205	249	266	249
279	279	282	289	264	263	227	227	279	284	270	227	270	284	270
279	279	279	279	352	336	326	315	304	293	281	270	270	284	270
279	279	279	279	343	327	316	305	294	284	271	257	270	284	270
279	279	279	279	324	307	296	285	279	284	270	229	270	284	270
279	279	279	279	315	298	286	275	279	284	270	227	270	284	270
279	279	279	279	305	288	276	264	279	284	270	227	270	284	270
311	314	321	328	236	263	227	227	279	284	270	227	270	284	270
281	284	292	299	250	263	227	227	279	284	270	227	270	284	270
291	294	302	309	236	263	227	227	279	284	270	227	270	284	270
301	304	312	319	236	263	227	227	279	284	270	227	270	284	270
321	324	331	338	236	263	227	227	279	284	270	227	270	284	270
330	333	340	347	236	263	227	227	279	284	270	227	270	284	270
339	342	349	356	236	263	227	227	279	284	270	227	270	284	270
279	279	279	279	362	345	335	324	314	303	291	280	270	284	270

75	**76**	**77**	**78**	**79**	**80**	**81**	**82**	**83**	**84**	**85**	**86**	**87**	**88**	**89**
113	**113**	**378**	**139**	**150**	**128**	**179**	**128**	**128**	**166**	**166**	**128**	**139**	**166**	**128**

89	90	91	92	93	94	95	96	97	98	99	100	101	102	103
320	314	294	312	314	314	321	345	294	314	314	294	312	294	314
320	314	294	312	314	314	314	345	294	314	314	294	312	294	314
320	314	294	312	314	314	314	345	294	314	314	294	312	294	314
320	314	294	312	314	314	314	348	310	314	314	294	312	294	314
299	310	321	332	342	353	363	316	262	283	283	262	280	270	288
291	303	314	325	335	346	356	316	262	283	283	262	280	262	283
284	295	306	318	328	339	349	300	240	266	266	240	263	250	273
277	288	299	310	321	332	342	300	240	266	266	240	263	240	266
272	280	291	303	314	325	335	300	240	266	266	240	263	240	266
289	283	284	295	306	318	328	316	262	283	283	262	280	262	283
272	266	277	288	299	310	321	300	240	266	266	240	263	240	266
289	283	269	280	291	303	314	316	262	283	283	262	280	262	283
272	266	259	273	284	295	306	300	240	266	266	240	263	240	266
289	283	262	280	283	288	299	316	262	283	283	262	280	262	283
272	266	240	263	269	280	291	300	240	266	266	240	263	240	266
289	283	262	280	283	283	284	316	262	283	283	262	280	262	283
272	266	240	263	266	266	277	300	240	266	266	240	263	240	266
289	283	262	280	283	283	283	316	262	283	283	262	280	262	283
272	266	240	263	266	266	266	300	240	266	266	240	263	240	266
289	283	262	280	283	283	283	316	262	283	283	262	280	262	283
289	283	262	280	283	283	283	316	262	283	283	262	280	262	283
272	266	240	263	266	266	266	310	270	266	266	240	263	240	266
289	283	262	280	283	283	283	318	278	283	283	262	280	262	283
272	266	240	263	266	266	266	325	285	274	266	240	263	240	266
289	283	262	280	283	283	283	332	293	283	283	262	280	262	283
272	266	240	263	266	266	266	339	300	289	266	250	263	240	266
289	283	262	280	283	283	283	346	308	296	283	262	280	262	283
272	266	240	263	266	266	266	360	322	311	289	278	266	250	266
289	283	262	280	283	283	283	366	329	319	296	285	280	262	283
272	266	240	263	266	266	266	372	336	326	304	293	282	270	266
272	266	240	263	266	266	266	378	343	333	311	300	289	278	266
289	283	262	280	283	283	283	385	350	340	319	308	296	285	283
299	310	321	332	342	353	363	241	172	193	223	240	255	270	288
291	303	314	325	335	346	356	258	181	211	213	229	245	261	280
284	295	306	318	328	339	349	241	162	193	201	218	234	250	273
282	293	304	315	326	336	347	258	181	211	211	215	231	247	270
277	288	299	310	321	332	342	241	162	193	193	207	224	240	264
271	282	293	304	315	326	336	173	128	137	179	198	215	231	256
269	280	291	303	314	325	335	241	162	193	193	196	213	229	254
259	273	284	295	306	318	328	241	162	193	193	183	202	218	243
256	271	282	293	304	316	326	241	162	193	193	179	198	215	240
248	264	277	288	299	310	321	220	128	170	170	172	189	207	232
241	257	271	282	293	305	316	220	137	170	170	163	180	199	225
238	254	269	280	291	303	314	220	141	170	170	159	177	196	222

227	243	259	273	284	295	306	222	158	170	170	141	167	183	211
225	241	257	271	283	294	305	224	162	170	170	138	166	180	209
216	232	248	264	277	288	299	232	172	170	170	128	166	172	200
209	225	241	257	272	283	294	239	179	170	170	127	166	165	191
205	222	238	254	269	280	291	243	183	193	193	162	189	162	193
220	211	227	243	259	273	284	258	196	211	211	181	208	181	211
220	211	226	242	258	272	283	258	197	211	211	181	208	181	211
202	200	216	232	248	264	277	264	207	193	193	162	189	162	193
220	211	210	226	242	258	272	270	214	211	211	181	208	181	211
202	193	205	222	238	254	269	273	218	202	193	162	189	162	193
220	211	194	211	227	243	259	280	229	213	211	181	208	181	211
202	193	181	200	216	232	248	288	240	224	193	172	189	162	193
220	211	181	208	213	229	245	290	243	227	211	181	208	181	211
202	193	170	189	205	222	238	295	250	234	202	183	189	162	193
220	211	181	208	211	213	229	301	259	243	211	193	208	181	211
202	193	162	189	194	211	227	303	261	245	213	196	189	162	193
220	211	181	208	211	211	216	310	270	256	224	207	208	181	211
202	193	162	189	193	197	214	312	272	258	226	210	193	175	193
272	266	240	263	266	266	266	318	278	266	266	240	263	240	266
220	211	181	208	211	211	211	323	283	272	242	226	210	192	211
272	266	240	263	266	266	266	325	285	274	266	240	263	240	266
202	193	162	189	193	193	193	332	293	282	256	240	224	207	193
220	211	181	208	211	211	211	333	294	283	258	242	226	209	211
202	193	162	189	193	193	193	339	300	289	266	250	234	218	193
181	170	139	166	170	170	170	343	305	294	272	257	241	225	200
185	170	143	166	170	170	170	346	308	296	274	261	245	229	203
198	179	162	166	170	170	170	353	315	304	282	270	256	240	215
199	180	164	166	170	170	170	354	316	305	283	271	257	241	216
209	191	174	166	170	170	170	360	322	311	289	278	266	250	225
216	199	180	166	170	170	170	364	326	316	293	282	271	257	232
220	203	185	169	170	170	170	366	329	319	296	285	274	261	236
231	215	198	179	170	170	170	372	336	326	304	293	282	270	247
241	225	209	191	174	170	170	378	343	333	311	300	289	278	257
305	316	327	337	348	358	368	166	181	200	232	248	264	277	294
288	299	310	321	332	342	353	196	149	172	207	223	239	255	277
277	288	299	310	321	332	342	165	150	149	189	207	224	240	264
264	277	288	299	310	321	332	181	150	128	172	189	207	224	248
248	264	277	288	299	310	321	220	128	170	170	172	189	207	232
232	248	264	277	288	299	310	227	149	177	177	149	174	189	216
216	232	248	264	277	288	299	232	172	154	154	128	149	172	200
200	216	232	248	264	277	288	248	189	172	128	150	128	150	181
181	200	216	232	248	264	277	264	207	189	154	128	147	128	165
187	181	200	216	232	248	264	277	224	207	177	149	174	136	177
166	154	172	189	207	224	240	294	248	232	200	181	165	139	154
0	128	150	172	189	207	224	305	264	248	216	200	181	165	128
128	0	132	149	172	189	207	316	277	264	232	216	200	181	149
150	132	0	128	149	172	189	327	288	277	248	232	216	200	172
172	149	128	0	128	149	172	337	299	288	264	248	232	216	189

189	172	149	128	0	128	149	348	310	299	277	264	248	232	207
207	189	172	149	128	0	128	358	321	310	288	277	264	248	224
224	207	189	172	149	128	0	368	332	321	299	288	277	264	240
305	316	327	337	348	358	368	0	196	200	232	248	264	277	294
264	277	288	299	310	321	332	196	0	132	172	189	207	224	248
248	264	277	288	299	310	321	200	132	0	149	172	189	207	232
216	232	248	264	277	288	299	232	172	149	0	132	149	172	200
200	216	232	248	264	277	288	248	189	172	132	0	128	149	181
181	200	216	232	248	264	277	264	207	189	149	128	0	128	165
165	181	200	216	232	248	264	277	224	207	172	149	128	0	139
128	149	172	189	207	224	240	294	248	232	200	181	165	139	0
99	128	149	172	189	207	224	305	264	248	216	200	181	165	128
149	128	128	128	149	172	189	327	288	277	248	232	216	200	172
172	149	128	128	132	149	172	337	299	288	264	248	232	216	189
189	172	149	128	70	128	149	348	310	299	277	264	248	232	207
207	189	172	149	128	70	128	358	321	310	288	277	264	248	224
224	207	189	172	149	128	70	368	332	321	299	288	277	264	240
175	166	196	170	166	166	169	340	301	290	268	252	236	220	194
225	209	196	174	166	166	166	369	333	322	300	289	278	266	241
256	270	282	293	304	315	326	191	239	211	211	239	215	239	240
218	234	250	266	278	289	300	231	239	211	211	239	215	239	211
207	224	240	256	270	282	293	241	239	211	211	239	215	239	211
202	211	239	215	229	245	261	279	239	211	211	239	215	239	211
202	211	239	215	211	218	234	298	254	238	211	239	215	239	211
227	236	263	240	236	236	236	321	282	270	240	263	240	263	236
227	236	263	240	236	236	236	369	333	322	300	289	278	266	241
280	274	251	271	274	274	274	385	350	340	319	308	296	285	274
277	284	304	286	284	284	284	391	357	347	326	315	304	304	284
277	284	304	286	284	284	284	279	304	284	284	304	286	304	284
277	284	304	286	284	284	284	378	343	333	311	304	289	304	284
277	284	304	286	284	284	284	372	336	326	304	304	286	304	284
277	284	304	286	284	284	284	366	329	319	296	304	286	304	284
277	284	304	286	284	284	284	360	322	311	289	304	286	304	284
277	284	304	286	284	284	284	353	315	304	284	304	286	304	284
277	284	304	286	284	284	284	346	308	296	284	304	286	304	284
277	284	304	286	284	284	284	339	304	289	284	304	286	304	284
277	284	304	286	284	284	284	325	304	284	284	304	286	304	284
277	284	304	286	284	284	284	318	304	284	284	304	286	304	284
277	284	304	286	284	284	284	310	304	284	284	304	286	304	284
277	284	304	286	284	284	284	295	304	284	284	304	286	304	284
277	284	304	286	284	284	284	288	304	284	284	304	286	304	284
277	284	304	286	284	284	284	385	350	340	319	308	296	304	284
277	284	304	289	300	311	322	246	304	284	284	304	286	304	284
300	311	322	333	343	354	364	246	304	284	284	304	286	304	289
293	304	315	326	336	347	357	246	304	284	284	304	286	304	284
285	296	308	319	329	340	350	246	304	284	284	304	286	304	284
278	289	304	311	322	333	343	246	304	284	284	304	286	304	284
277	284	304	286	284	284	284	272	304	284	284	304	286	304	284

277	284	304	296	308	319	329	246	304	284	284	304	286	304	284
277	284	304	304	315	326	336	246	304	284	284	304	286	304	284
277	284	304	286	293	304	315	246	304	284	284	304	286	304	284
277	284	304	286	285	296	308	246	304	284	284	304	286	304	284
277	284	304	286	284	289	300	246	304	284	284	304	286	304	284
277	284	304	286	284	284	293	246	304	284	284	304	286	304	284
198	215	231	247	263	275	287	250	196	174	166	196	170	196	179
185	204	220	236	252	268	279	261	203	185	166	196	170	196	169
257	241	236	211	208	208	208	388	354	343	322	311	300	289	271
239	223	236	211	208	208	208	377	342	331	310	299	288	276	255
222	211	239	215	211	211	211	367	330	320	298	286	275	262	238
202	211	239	215	211	211	211	347	309	298	275	262	246	239	211
202	211	239	215	211	211	211	337	299	288	264	248	232	239	211
202	211	239	215	211	211	211	328	289	278	250	239	218	239	211
202	211	239	215	211	211	211	319	279	268	236	239	215	239	211
202	211	239	215	211	217	233	299	255	239	211	239	215	239	211
202	211	239	215	215	231	247	289	241	225	211	239	215	239	211
202	211	239	224	240	256	270	271	239	211	211	239	215	239	211
204	220	239	252	268	279	290	245	239	211	211	239	215	239	211
233	249	265	277	288	299	310	216	239	211	211	239	215	239	217
202	211	239	215	211	211	218	309	269	254	222	239	215	239	211
227	236	263	240	249	265	277	264	263	236	236	263	240	263	236
227	236	263	240	236	236	236	352	315	304	281	270	255	263	236
227	236	263	240	236	236	236	343	305	294	271	263	241	263	236
227	236	263	240	236	236	236	324	285	274	245	263	240	263	236
227	236	263	240	236	236	236	315	275	262	236	263	240	263	236
227	236	263	240	236	236	236	305	264	248	236	263	240	263	236
241	257	272	283	294	305	316	207	263	236	236	263	240	263	236
227	236	263	247	263	275	287	250	263	236	236	263	240	263	236
227	236	263	261	274	285	297	236	263	236	236	263	240	263	236
227	243	263	273	284	295	307	221	263	236	236	263	240	263	236
256	270	282	293	304	315	326	191	263	236	236	263	240	263	240
269	280	292	303	314	325	335	187	263	236	236	263	240	263	254
279	290	302	313	324	334	345	187	263	236	236	263	240	263	268
227	236	263	240	236	236	236	362	324	314	291	280	269	263	236

90	**91**	**92**	**93**	**94**	**95**	**96**	**97**	**98**	**99**	**100**	**101**	**102**	**103**	**104**
132	**128**	**128**	**128**	**128**	**368**	**196**	**132**	**149**	**132**	**128**	**128**	**139**	**128**	**149**

104	105	106	107	108	109	110	111	112	113	114	115	116	117	118
314	312	294	314	314	321	345	345	370	370	370	370	370	384	384
314	312	294	314	314	314	345	345	370	370	370	370	370	384	384
314	312	294	314	314	314	345	345	370	370	370	370	370	384	384
314	312	294	314	314	314	345	345	370	370	370	370	370	384	384
299	321	332	342	353	363	334	364	344	344	344	344	344	359	364
291	314	325	335	346	356	327	357	344	344	344	344	344	359	359
284	306	318	328	339	349	320	350	329	329	329	329	329	345	350
277	299	310	321	332	342	313	343	329	329	329	329	329	345	345
269	291	303	314	325	335	305	336	329	329	329	329	329	345	345
283	284	295	306	318	328	316	329	344	344	344	344	344	359	359
266	277	288	299	310	321	300	322	329	329	329	329	329	345	345
283	280	280	291	303	314	316	316	344	344	344	344	344	359	359
266	263	273	284	295	306	300	308	329	329	329	329	329	345	345
283	280	264	283	288	299	316	316	344	344	344	344	344	359	359
266	263	254	269	280	291	300	300	329	329	329	329	329	345	345
283	280	262	283	283	284	316	316	344	344	344	344	344	359	359
266	263	240	266	266	277	300	300	329	329	329	329	329	345	345
283	280	262	283	283	283	316	316	344	344	344	344	344	359	359
266	263	240	266	266	266	300	300	329	329	329	329	329	345	345
283	280	262	283	283	283	316	316	344	344	344	344	344	359	359
283	280	262	283	283	283	316	316	344	344	344	344	344	359	359
266	263	240	266	266	266	300	300	329	329	329	329	329	345	345
283	280	262	283	283	283	316	316	344	344	344	344	344	359	359
266	263	240	266	266	266	300	300	329	329	329	329	329	345	345
283	280	262	283	283	283	316	316	344	344	344	344	344	359	359
266	263	240	266	266	266	300	300	329	329	329	329	329	345	345
283	280	262	283	283	283	316	316	344	344	344	344	344	359	359
266	263	240	266	266	266	300	300	329	329	329	329	329	345	345
283	280	262	283	283	283	316	316	344	344	344	344	344	359	359
266	263	240	266	266	266	300	300	330	329	329	329	329	345	345
266	263	240	266	266	266	300	300	337	329	329	329	329	345	345
283	280	262	283	283	283	316	316	344	344	344	344	344	359	359
299	321	332	342	353	363	334	364	276	276	276	276	291	315	364
291	314	325	335	346	356	327	357	289	289	289	289	289	308	357
284	306	318	328	339	349	320	350	276	276	276	276	277	300	350
282	304	315	326	336	347	318	348	289	289	289	289	289	307	348
277	299	310	321	332	342	313	343	276	276	276	276	276	294	343
271	293	304	315	326	336	307	338	215	215	215	235	261	287	338
269	291	303	314	325	335	305	336	276	276	276	276	276	294	336
259	284	295	306	318	328	298	329	276	276	276	276	276	294	329
256	282	293	304	316	326	296	327	276	276	276	276	276	294	327
248	277	288	299	310	321	290	322	261	261	261	261	261	279	322
241	271	282	293	305	316	285	317	261	261	261	261	261	279	317
238	269	280	291	303	314	283	315	261	261	261	261	261	279	315

227	259	273	284	295	306	275	308	261	261	261	261	261	279	308
225	257	271	283	294	305	274	306	261	261	261	261	261	279	306
216	248	264	277	288	299	268	300	261	261	261	261	261	279	300
209	241	257	272	283	294	261	295	261	261	261	261	261	279	295
205	238	254	269	280	291	257	293	276	276	276	276	276	294	294
211	227	243	259	273	284	258	285	289	289	289	289	289	307	307
211	226	242	258	272	283	258	284	289	289	289	289	289	307	307
193	216	232	248	264	277	241	278	276	276	276	276	276	294	294
211	210	226	242	258	272	258	273	289	289	289	289	289	307	307
193	205	222	238	254	269	241	270	276	276	276	276	276	294	294
211	208	211	227	243	259	258	261	289	289	289	289	289	307	307
193	189	200	216	232	248	241	250	276	276	276	276	276	294	294
211	208	196	213	229	245	258	258	289	289	289	289	289	307	307
193	189	187	205	222	238	241	241	276	276	276	276	276	294	294
211	208	181	211	213	229	258	258	289	289	289	289	289	307	307
193	189	175	194	211	227	241	241	276	276	276	276	276	294	294
211	208	181	211	211	216	258	258	289	289	289	289	289	307	307
193	189	162	193	197	214	241	241	276	276	276	276	276	294	294
266	263	240	266	266	266	300	300	329	329	329	329	329	345	345
211	208	181	211	211	211	258	258	289	289	289	289	289	307	307
266	263	240	266	266	266	300	300	329	329	329	329	329	345	345
193	189	162	193	193	193	241	241	286	276	276	276	276	294	294
211	208	181	211	211	211	258	258	289	289	289	289	289	307	307
193	189	162	193	193	193	241	241	294	276	276	276	276	294	294
181	166	127	170	170	170	220	220	299	273	264	261	261	279	279
185	166	127	170	170	170	220	220	301	275	268	261	261	279	279
198	166	137	170	170	170	220	220	309	283	275	261	261	279	279
199	166	138	170	170	170	220	220	310	284	276	261	261	279	279
209	174	152	170	170	170	220	220	316	290	283	261	261	279	279
216	180	163	170	170	170	220	220	321	295	287	261	261	279	279
220	185	169	170	170	170	220	220	323	298	290	261	261	279	279
231	198	179	170	170	170	220	220	330	305	298	268	261	279	279
241	209	191	174	170	170	220	220	337	313	305	275	261	279	279
305	327	337	348	358	368	340	369	211	231	241	279	298	321	369
288	310	321	332	342	353	323	354	239	239	239	259	280	304	354
277	299	310	321	332	342	313	343	202	202	202	243	269	293	343
264	288	299	310	321	332	301	333	202	202	202	227	254	282	333
248	277	288	299	310	321	290	322	261	261	261	261	261	279	322
232	264	277	288	299	310	279	311	266	266	266	266	266	284	311
216	248	264	277	288	299	268	300	249	249	249	249	249	270	300
200	232	248	264	277	288	252	289	202	202	202	202	202	227	289
181	216	232	248	264	277	236	278	249	249	249	249	249	270	278
177	200	216	232	248	264	227	266	266	266	266	266	266	284	284
154	172	189	207	224	240	208	241	249	249	249	249	249	270	270
99	149	172	189	207	224	175	225	256	218	207	202	202	227	227
128	128	149	172	189	207	166	209	270	234	224	211	211	236	236
149	128	128	149	172	189	196	196	282	250	240	239	239	263	263
172	128	128	128	149	172	170	174	293	266	256	215	215	240	240

189	149	132	70	128	149	166	166	304	278	270	229	211	236	236
207	172	149	128	70	128	166	166	315	289	282	245	218	236	236
224	189	172	149	128	70	169	166	326	300	293	261	234	236	236
305	327	337	348	358	368	340	369	191	231	241	279	298	321	369
264	288	299	310	321	332	301	333	239	239	239	239	254	282	333
248	277	288	299	310	321	290	322	211	211	211	211	238	270	322
216	248	264	277	288	299	268	300	211	211	211	211	211	240	300
200	232	248	264	277	288	252	289	239	239	239	239	239	263	289
181	216	232	248	264	277	236	278	215	215	215	215	215	240	278
165	200	216	232	248	264	220	266	239	239	239	239	239	263	266
128	172	189	207	224	240	194	241	240	211	211	211	211	236	241
0	149	172	189	207	224	175	225	256	218	211	211	211	236	236
149	0	128	149	172	189	170	191	282	250	240	215	215	240	240
172	128	0	132	149	172	196	196	293	266	256	239	239	263	263
189	149	132	0	128	149	166	166	304	278	270	229	211	236	236
207	172	149	128	0	128	166	166	315	289	282	245	218	236	236
224	189	172	149	128	0	169	166	326	300	293	261	234	236	236
175	170	196	166	166	169	0	170	295	269	259	216	187	187	187
225	191	196	166	166	166	170	0	327	301	294	263	236	202	187
256	282	293	304	315	326	295	327	0	159	172	218	245	275	327
218	250	266	278	289	300	269	301	159	0	113	177	207	241	301
211	240	256	270	282	293	259	294	172	113	0	167	196	231	294
211	215	239	229	245	261	216	263	218	177	167	0	141	185	263
211	215	239	211	218	234	187	236	245	207	196	141	0	152	236
236	240	263	236	236	236	187	202	275	241	231	185	152	0	202
236	240	263	236	236	236	187	187	327	301	294	263	236	202	0
274	271	251	274	274	274	308	308	344	336	336	336	336	352	352
284	286	304	284	284	284	246	246	351	327	320	290	272	240	177
284	286	304	284	284	284	246	263	218	204	204	204	204	185	263
284	286	304	284	284	284	246	246	337	313	305	275	252	218	177
284	286	304	284	284	284	246	246	330	305	298	268	241	207	177
284	286	304	284	284	284	246	246	323	298	290	257	231	196	177
284	286	304	284	284	284	246	246	316	290	283	247	220	183	177
284	286	304	284	284	284	246	246	309	283	275	236	209	177	177
284	286	304	284	284	284	246	246	301	275	268	225	204	177	177
284	286	304	284	284	284	246	246	294	268	257	215	204	177	177
284	286	304	284	284	284	246	246	279	247	236	204	204	177	196
284	286	304	284	284	284	246	246	272	236	225	204	204	177	207
284	286	304	284	284	284	246	246	263	225	215	204	204	177	218
284	286	304	284	284	284	246	246	241	204	204	204	204	177	240
284	286	304	284	284	284	246	250	231	204	204	204	204	177	250
284	286	304	284	284	284	246	246	344	320	313	283	263	229	177
284	286	304	300	311	322	291	323	204	204	204	213	240	272	323
300	322	333	343	354	364	335	365	204	223	234	274	293	316	365
293	315	326	336	347	357	328	358	204	213	223	266	285	309	358
285	308	319	329	340	350	321	351	204	204	213	255	278	301	351
284	300	311	322	333	343	314	344	204	204	204	245	270	294	344
284	286	304	284	284	284	246	272	207	204	204	204	204	198	272

284	286	304	308	319	329	299	330	204	204	204	224	250	279	330
284	293	304	315	326	336	306	337	204	204	204	234	261	286	337
284	286	304	293	304	315	284	316	204	204	204	204	229	263	316
284	286	304	285	296	308	277	309	204	204	204	204	218	252	309
284	286	304	284	289	300	269	301	204	204	204	204	207	241	301
284	286	304	284	284	293	259	294	204	204	204	204	204	231	294
198	231	247	263	275	287	250	288	180	160	160	160	185	222	288
185	220	236	252	268	279	240	280	193	160	160	160	174	211	280
257	225	236	208	208	208	205	154	348	323	316	286	268	234	149
239	211	236	208	208	208	185	154	336	311	304	274	250	216	133
222	215	239	211	211	211	167	160	324	299	291	259	232	197	129
211	215	239	211	211	211	160	160	303	276	269	227	199	161	156
211	215	239	211	211	211	160	174	293	266	255	213	182	138	174
211	215	239	211	211	211	160	189	283	252	241	197	169	129	189
211	215	239	211	211	211	160	206	273	238	227	180	146	129	206
211	215	239	211	217	233	185	234	246	209	197	143	71	150	234
211	215	239	215	231	247	202	249	232	193	180	126	120	171	249
211	215	239	240	256	270	227	272	207	167	149	113	159	198	272
211	236	252	268	279	290	256	292	175	126	83	162	192	227	292
233	265	277	288	299	310	279	312	136	126	139	194	222	256	312
211	215	239	211	211	218	171	220	261	223	213	167	128	130	220
236	240	263	249	265	277	236	278	197	152	136	132	171	207	278
236	240	263	236	236	236	187	187	309	283	275	236	209	172	141
236	240	263	236	236	236	187	187	299	273	264	222	193	152	165
236	240	263	236	236	236	187	196	279	246	236	191	161	90	196
236	240	263	236	236	236	187	211	269	232	222	175	138	107	211
236	240	263	236	236	236	187	225	255	218	207	158	129	137	225
241	272	283	294	305	316	285	318	129	137	153	204	231	265	318
236	240	263	263	275	287	250	288	180	132	129	153	185	222	288
236	245	263	274	285	297	265	298	167	129	129	172	202	236	298
236	259	273	284	295	307	275	308	143	129	133	187	217	250	308
256	282	293	304	315	326	295	327	129	159	172	218	245	275	327
269	292	303	314	325	335	305	336	129	176	187	233	259	285	336
279	302	313	324	334	345	315	346	144	192	204	247	272	295	346
236	240	263	236	236	236	187	187	319	293	285	250	223	187	120

105	**106**	**107**	**108**	**109**	**110**	**111**	**112**	**113**	**114**	**115**	**116**	**117**	**118**	**119**
128	**132**	**128**	**128**	**169**	**170**	**327**	**159**	**113**	**167**	**141**	**152**	**202**	**352**	**385**

119	120	121	122	123	124	125	126	127	128	129	130	131	132	133
340	416	416	416	416	416	416	416	416	416	416	416	416	416	416
333	416	416	416	416	416	416	416	416	416	416	416	416	416	416
296	416	416	416	416	416	416	416	416	416	416	416	416	416	416
185	416	416	416	416	416	416	416	416	416	416	416	416	416	416
379	391	391	391	391	391	391	391	391	391	391	391	391	391	391
373	391	391	391	391	391	391	391	391	391	391	391	391	391	391
367	378	378	378	378	378	378	378	378	378	378	378	378	378	378
361	378	378	378	378	378	378	378	378	378	378	378	378	378	378
354	378	378	378	378	378	378	378	378	378	378	378	378	378	378
347	391	391	391	391	391	391	391	391	391	391	391	391	391	391
340	378	378	378	378	378	378	378	378	378	378	378	378	378	378
333	391	391	391	391	391	391	391	391	391	391	391	391	391	391
326	378	378	378	378	378	378	378	378	378	378	378	378	378	378
319	391	391	391	391	391	391	391	391	391	391	391	391	391	391
311	378	378	378	378	378	378	378	378	378	378	378	378	378	378
304	391	391	391	391	391	391	391	391	391	391	391	391	391	391
296	378	378	378	378	378	378	378	378	378	378	378	378	378	378
289	391	391	391	391	391	391	391	391	391	391	391	391	391	391
282	378	378	378	378	378	378	378	378	378	378	378	378	378	378
266	391	391	391	391	391	391	391	391	391	391	391	391	391	391
256	391	391	391	391	391	391	391	391	391	391	391	391	391	391
245	378	378	378	378	378	378	378	378	378	378	378	378	378	378
234	391	391	391	391	391	391	391	391	391	391	391	391	391	391
224	378	378	378	378	378	378	378	378	378	378	378	378	378	378
213	391	391	391	391	391	391	391	391	391	391	391	391	391	391
202	378	378	378	378	378	378	378	378	378	378	378	378	378	378
189	391	391	391	391	391	391	391	391	391	391	391	391	391	391
167	378	378	378	378	378	378	378	378	378	378	378	378	378	378
149	391	391	391	391	391	391	391	391	391	391	391	391	391	391
134	378	378	378	378	378	378	378	378	378	378	378	378	378	378
113	378	378	378	378	378	378	378	378	378	378	378	378	378	378
111	391	391	391	391	391	391	391	391	391	391	391	391	391	391
379	386	333	373	367	361	354	347	340	333	333	333	333	333	333
373	379	345	367	361	354	347	345	345	345	345	345	345	345	345
367	373	333	361	354	347	340	333	333	333	333	333	333	333	333
365	371	345	358	351	345	345	345	345	345	345	345	345	345	345
361	367	333	354	347	340	333	333	333	333	333	333	333	333	333
355	362	286	348	341	334	327	320	313	305	290	286	286	286	286
354	361	333	347	340	333	333	333	333	333	333	333	333	333	333
347	354	333	340	333	333	333	333	333	333	333	333	333	333	333
345	352	333	338	333	333	333	333	333	333	333	333	333	333	333
340	347	320	333	326	320	320	320	320	320	320	320	320	320	320
335	342	320	328	321	320	320	320	320	320	320	320	320	320	320
333	340	320	326	320	320	320	320	320	320	320	320	320	320	320

326	333	320	320	320	320	320	320	320	320	320	320	320	320	320
324	331	320	320	320	320	320	320	320	320	320	320	320	320	320
319	326	320	320	320	320	320	320	320	320	320	320	320	320	320
314	321	320	320	320	320	320	320	320	320	320	320	320	320	320
311	333	333	333	333	333	333	333	333	333	333	333	333	333	333
304	345	345	345	345	345	345	345	345	345	345	345	345	345	345
303	345	345	345	345	345	345	345	345	345	345	345	345	345	345
296	333	333	333	333	333	333	333	333	333	333	333	333	333	333
292	345	345	345	345	345	345	345	345	345	345	345	345	345	345
289	333	333	333	333	333	333	333	333	333	333	333	333	333	333
282	345	345	345	345	345	345	345	345	345	345	345	345	345	345
274	333	333	333	333	333	333	333	333	333	333	333	333	333	333
272	345	345	345	345	345	345	345	345	345	345	345	345	345	345
266	333	333	333	333	333	333	333	333	333	333	333	333	333	333
258	345	345	345	345	345	345	345	345	345	345	345	345	345	345
256	333	333	333	333	333	333	333	333	333	333	333	333	333	333
245	345	345	345	345	345	345	345	345	345	345	345	345	345	345
242	333	333	333	333	333	333	333	333	333	333	333	333	333	333
234	378	378	378	378	378	378	378	378	378	378	378	378	378	378
226	345	345	345	345	345	345	345	345	345	345	345	345	345	345
224	378	378	378	378	378	378	378	378	378	378	378	378	378	378
213	333	333	333	333	333	333	333	333	333	333	333	333	333	333
211	345	345	345	345	345	345	345	345	345	345	345	345	345	345
208	333	333	333	333	333	333	333	333	333	333	333	333	333	333
229	320	320	320	320	320	320	320	320	320	320	320	320	320	320
229	320	320	320	320	320	320	320	320	320	320	320	320	320	320
229	320	320	320	320	320	320	320	320	320	320	320	320	320	320
229	320	320	320	320	320	320	320	320	320	320	320	320	320	320
229	320	320	320	320	320	320	320	320	320	320	320	320	320	320
229	320	320	320	320	320	320	320	320	320	320	320	320	320	320
229	320	320	320	320	320	320	320	320	320	320	320	320	320	320
229	320	320	320	320	320	320	320	320	320	320	320	320	320	320
229	320	320	320	320	320	320	320	320	320	320	320	320	320	320
385	391	284	378	372	366	360	353	346	339	325	318	310	295	288
370	376	304	364	357	350	343	336	329	322	308	304	304	304	304
361	367	277	354	347	340	333	326	319	311	296	289	282	277	277
350	357	277	343	336	329	322	315	308	300	285	278	277	277	277
340	347	320	333	326	320	320	320	320	320	320	320	320	320	320
329	336	324	324	324	324	324	324	324	324	324	324	324	324	324
319	326	312	312	312	312	312	312	312	312	312	312	312	312	312
308	315	277	300	293	285	278	277	277	277	277	277	277	277	277
296	312	312	312	312	312	312	312	312	312	312	312	312	312	312
285	324	324	324	324	324	324	324	324	324	324	324	324	324	324
268	312	312	312	312	312	312	312	312	312	312	312	312	312	312
280	277	277	277	277	277	277	277	277	277	277	277	277	277	277
274	284	284	284	284	284	284	284	284	284	284	284	284	284	284
251	304	304	304	304	304	304	304	304	304	304	304	304	304	304
271	286	286	286	286	286	286	286	286	286	286	286	286	286	286

274	284	284	284	284	284	284	284	284	284	284	284	284	284	284
274	284	284	284	284	284	284	284	284	284	284	284	284	284	284
274	284	284	284	284	284	284	284	284	284	284	284	284	284	284
385	391	279	378	372	366	360	353	346	339	325	318	310	295	288
350	357	304	343	336	329	322	315	308	304	304	304	304	304	304
340	347	284	333	326	319	311	304	296	289	284	284	284	284	284
319	326	284	311	304	296	289	284	284	284	284	284	284	284	284
308	315	304	304	304	304	304	304	304	304	304	304	304	304	304
296	304	286	289	286	286	286	286	286	286	286	286	286	286	286
285	304	304	304	304	304	304	304	304	304	304	304	304	304	304
274	284	284	284	284	284	284	284	284	284	284	284	284	284	284
274	284	284	284	284	284	284	284	284	284	284	284	284	284	284
271	286	286	286	286	286	286	286	286	286	286	286	286	286	286
251	304	304	304	304	304	304	304	304	304	304	304	304	304	304
274	284	284	284	284	284	284	284	284	284	284	284	284	284	284
274	284	284	284	284	284	284	284	284	284	284	284	284	284	284
274	284	284	284	284	284	284	284	284	284	284	284	284	284	284
308	246	246	246	246	246	246	246	246	246	246	246	246	246	246
308	246	263	246	246	246	246	246	246	246	246	246	246	246	250
344	351	218	337	330	323	316	309	301	294	279	272	263	241	231
336	327	204	313	305	298	290	283	275	268	247	236	225	204	204
336	320	204	305	298	290	283	275	268	257	236	225	215	204	204
336	290	204	275	268	257	247	236	225	215	204	204	204	204	204
336	272	204	252	241	231	220	209	204	204	204	204	204	204	204
352	240	185	218	207	196	183	177	177	177	177	177	177	177	177
352	177	263	177	177	177	177	177	177	177	196	207	218	240	250
0	385	385	385	385	385	385	385	385	385	385	385	385	385	385
385	0	290	134	149	167	177	189	202	213	234	245	256	274	282
385	290	0	275	268	257	247	236	225	215	191	179	169	137	119
385	134	275	0	113	134	149	167	177	189	213	224	234	256	266
385	149	268	113	0	113	134	149	167	177	202	213	224	245	256
385	167	257	134	113	0	113	134	149	167	189	202	213	234	245
385	177	247	149	134	113	0	113	134	149	177	189	202	224	234
385	189	236	167	149	134	113	0	113	134	167	177	189	213	224
385	202	225	177	167	149	134	113	0	113	149	167	177	202	213
385	213	215	189	177	167	149	134	113	0	134	149	167	189	202
385	234	191	213	202	189	177	167	149	134	0	113	134	167	177
385	245	179	224	213	202	189	177	167	149	113	0	113	149	167
385	256	169	234	224	213	202	189	177	167	134	113	0	134	149
385	274	137	256	245	234	224	213	202	189	167	149	134	0	113
385	282	119	266	256	245	234	224	213	202	177	167	149	113	0
385	113	283	113	134	149	167	177	189	202	224	234	245	266	274
385	348	213	334	327	320	313	305	298	290	275	268	257	236	225
385	387	274	374	368	362	355	348	341	334	320	313	305	290	283
385	380	266	368	362	355	348	341	334	327	313	305	298	283	275
385	374	255	362	355	348	341	334	327	320	305	298	290	275	268
385	368	245	355	348	341	334	327	320	313	298	290	283	268	257
385	298	113	283	275	268	257	247	236	225	203	191	179	152	137

385	355	224	341	334	327	320	313	305	298	283	275	268	247	236
385	362	234	348	341	334	327	320	313	305	290	283	275	257	247
385	341	202	327	320	313	305	298	290	283	268	257	247	225	215
385	334	189	320	313	305	298	290	283	275	257	247	236	215	203
385	327	177	313	305	298	290	283	275	268	247	236	225	203	191
385	320	167	305	298	290	283	275	268	257	236	225	215	191	179
308	314	246	299	292	284	277	269	259	249	246	246	246	246	246
308	307	246	292	284	277	269	259	249	246	246	246	246	246	246
333	207	286	207	207	207	207	207	207	207	229	239	250	270	278
333	207	274	207	207	207	207	207	207	207	211	222	232	254	264
336	204	259	204	204	204	204	204	204	204	204	204	214	236	246
336	204	227	204	204	204	204	204	204	204	204	204	204	204	214
336	215	213	204	204	204	204	204	204	204	204	204	204	204	204
336	229	204	207	204	204	204	204	204	204	204	204	204	204	204
336	243	204	222	211	204	204	204	204	204	204	204	204	204	204
336	270	204	250	240	229	218	207	204	204	204	204	204	204	204
336	280	204	265	254	243	233	222	211	204	204	204	204	204	204
336	298	204	283	275	268	257	247	236	225	204	204	204	204	204
336	318	204	303	295	288	280	273	265	254	233	222	211	204	204
336	336	204	322	315	308	300	293	285	278	261	250	240	218	207
336	258	204	236	225	215	204	204	204	204	204	204	204	204	204
352	304	177	289	282	274	266	256	245	234	213	202	189	177	177
352	189	236	177	177	177	177	177	177	177	177	177	189	213	223
352	206	222	181	177	177	177	177	177	177	177	177	177	197	209
352	234	191	213	202	189	177	177	177	177	177	177	177	177	177
352	249	177	227	217	206	194	181	177	177	177	177	177	177	177
352	263	177	241	231	220	209	198	185	177	177	177	177	177	177
352	342	204	328	321	314	307	299	292	284	269	259	249	227	217
352	314	177	299	292	284	277	269	259	249	227	217	206	181	177
352	324	177	309	302	294	287	279	272	263	241	231	220	198	185
352	333	187	319	312	304	297	289	282	274	256	245	234	213	202
352	352	218	338	331	324	316	309	302	294	279	272	263	241	231
354	361	233	347	340	333	326	319	312	304	289	282	274	256	245
363	369	247	356	349	342	335	328	321	314	299	292	284	269	259
352	177	250	177	177	177	177	177	177	177	180	193	205	227	238

120	**121**	**122**	**123**	**124**	**125**	**126**	**127**	**128**	**129**	**130**	**131**	**132**	**133**	**134**
290	**275**	**113**	**113**	**113**	**113**	**113**	**113**	**134**	**113**	**113**	**134**	**113**	**274**	**341**

134	135	136	137	138	139	140	141	142	143	144	145	146	147	148
416	416	416	416	416	416	416	416	416	416	416	416	416	345	345
416	416	416	416	416	416	416	416	416	416	416	416	416	345	345
416	416	416	416	416	416	416	416	416	416	416	416	416	345	345
416	416	416	416	416	416	416	416	416	416	416	416	416	345	345
391	391	391	391	391	391	391	391	391	391	391	391	391	316	316
391	391	391	391	391	391	391	391	391	391	391	391	391	316	316
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
391	391	391	391	391	391	391	391	391	391	391	391	391	316	316
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
391	391	391	391	391	391	391	391	391	391	391	391	391	316	316
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
391	391	391	391	391	391	391	391	391	391	391	391	391	316	316
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
391	391	391	391	391	391	391	391	391	391	391	391	391	316	316
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
391	391	391	391	391	391	391	391	391	391	391	391	391	316	316
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
391	391	391	391	391	391	391	391	391	391	391	391	391	316	316
391	391	391	391	391	391	391	391	391	391	391	391	391	316	316
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
391	391	391	391	391	391	391	391	391	391	391	391	391	316	316
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
391	391	391	391	391	391	391	391	391	391	391	391	391	316	316
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
391	391	391	391	391	391	391	391	391	391	391	391	391	316	316
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
391	391	391	391	391	391	391	391	391	391	391	391	391	316	316
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
391	391	391	391	391	391	391	391	391	391	391	391	391	316	316
379	333	333	333	333	333	333	333	333	333	333	333	333	241	252
373	345	345	345	345	345	345	345	345	345	345	345	345	258	258
367	333	333	333	333	333	333	333	333	333	333	333	333	241	241
365	345	345	345	345	345	345	345	345	345	345	345	345	258	258
361	333	333	333	333	333	333	333	333	333	333	333	333	241	241
355	286	286	286	286	286	286	286	286	286	286	286	286	200	211
354	333	333	333	333	333	333	333	333	333	333	333	333	241	241
347	333	333	333	333	333	333	333	333	333	333	333	333	241	241
345	333	333	333	333	333	333	333	333	333	333	333	333	241	241
340	320	320	320	320	320	320	320	320	320	320	320	320	220	220
335	320	320	320	320	320	320	320	320	320	320	320	320	220	220
333	320	320	320	320	320	320	320	320	320	320	320	320	220	220

326	320	320	320	320	320	320	320	320	320	320	320	320	220	220
324	320	320	320	320	320	320	320	320	320	320	320	320	220	220
320	320	320	320	320	320	320	320	320	320	320	320	320	220	220
320	320	320	320	320	320	320	320	320	320	320	320	320	220	220
333	333	333	333	333	333	333	333	333	333	333	333	333	241	241
345	345	345	345	345	345	345	345	345	345	345	345	345	258	258
345	345	345	345	345	345	345	345	345	345	345	345	345	258	258
333	333	333	333	333	333	333	333	333	333	333	333	333	241	241
345	345	345	345	345	345	345	345	345	345	345	345	345	258	258
333	333	333	333	333	333	333	333	333	333	333	333	333	241	241
345	345	345	345	345	345	345	345	345	345	345	345	345	258	258
333	333	333	333	333	333	333	333	333	333	333	333	333	241	241
345	345	345	345	345	345	345	345	345	345	345	345	345	258	258
333	333	333	333	333	333	333	333	333	333	333	333	333	241	241
345	345	345	345	345	345	345	345	345	345	345	345	345	258	258
333	333	333	333	333	333	333	333	333	333	333	333	333	241	241
345	345	345	345	345	345	345	345	345	345	345	345	345	258	258
333	333	333	333	333	333	333	333	333	333	333	333	333	241	241
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
345	345	345	345	345	345	345	345	345	345	345	345	345	258	258
378	378	378	378	378	378	378	378	378	378	378	378	378	300	300
333	333	333	333	333	333	333	333	333	333	333	333	333	241	241
345	345	345	345	345	345	345	345	345	345	345	345	345	258	258
333	333	334	333	333	333	333	333	333	333	333	333	333	249	241
320	320	339	332	325	320	320	320	320	320	320	320	320	256	245
320	320	341	334	327	320	320	320	320	320	320	320	320	259	249
320	320	348	341	334	327	320	320	320	320	320	320	320	269	259
320	320	349	342	335	328	320	320	321	320	320	320	320	270	261
320	320	355	348	341	334	320	320	327	320	320	320	320	277	269
320	320	359	352	345	338	320	324	331	320	320	320	320	281	274
320	320	362	355	348	341	320	327	334	320	320	320	320	284	277
320	327	368	362	355	348	320	334	341	320	320	320	320	292	284
320	334	374	368	362	355	320	341	348	327	320	320	320	299	292
385	284	284	284	284	284	284	284	284	284	284	284	284	250	261
370	304	304	304	304	304	304	304	304	304	304	304	304	225	236
361	277	277	277	277	277	277	277	277	277	277	277	277	209	220
350	277	277	277	277	277	277	277	277	277	277	277	277	191	203
340	320	320	320	320	320	320	320	320	320	320	320	320	220	220
329	324	324	324	324	324	324	324	324	324	324	324	324	227	227
319	312	312	312	312	312	312	312	312	312	312	312	312	208	208
308	277	277	277	277	277	277	277	277	277	277	277	277	154	154
312	312	312	312	312	312	312	312	312	312	312	312	312	208	208
324	324	324	324	324	324	324	324	324	324	324	324	324	227	227
312	312	312	312	312	312	312	312	312	312	312	312	312	208	208
277	277	300	293	285	278	277	277	277	277	277	277	277	198	185
284	284	311	304	296	289	284	284	284	284	284	284	284	215	204
304	304	322	315	308	304	304	304	304	304	304	304	304	231	220
286	289	333	326	319	311	286	296	304	286	286	286	286	247	236

284	300	343	336	329	322	284	308	315	293	285	284	284	263	252
284	311	354	347	340	333	284	319	326	304	296	289	284	275	268
284	322	364	357	350	343	284	329	336	315	308	300	293	287	279
385	246	246	246	246	246	272	246	246	246	246	246	246	250	261
350	304	304	304	304	304	304	304	304	304	304	304	304	196	203
340	284	284	284	284	284	284	284	284	284	284	284	284	174	185
319	284	284	284	284	284	284	284	284	284	284	284	284	166	166
308	304	304	304	304	304	304	304	304	304	304	304	304	196	196
296	286	286	286	286	286	286	286	286	286	286	286	286	170	170
304	304	304	304	304	304	304	304	304	304	304	304	304	196	196
284	284	289	284	284	284	284	284	284	284	284	284	284	179	169
284	284	300	293	285	284	284	284	284	284	284	284	284	198	185
286	286	322	315	308	300	286	286	293	286	286	286	286	231	220
304	304	333	326	319	311	304	304	304	304	304	304	304	247	236
284	300	343	336	329	322	284	308	315	293	285	284	284	263	252
284	311	354	347	340	333	284	319	326	304	296	289	284	275	268
284	322	364	357	350	343	284	329	336	315	308	300	293	287	279
246	291	335	328	321	314	246	299	306	284	277	269	259	250	240
246	323	365	358	351	344	272	330	337	316	309	301	294	288	280
344	204	204	204	204	204	207	204	204	204	204	204	204	180	193
320	204	223	213	204	204	204	204	204	204	204	204	204	160	160
313	204	234	223	213	204	204	204	204	204	204	204	204	160	160
283	213	274	266	255	245	204	224	234	204	204	204	204	160	160
263	240	293	285	278	270	204	250	261	229	218	207	204	185	174
229	272	316	309	301	294	198	279	286	263	252	241	231	222	211
177	323	365	358	351	344	272	330	337	316	309	301	294	288	280
385	385	385	385	385	385	385	385	385	385	385	385	385	308	308
113	348	387	380	374	368	298	355	362	341	334	327	320	314	307
283	213	274	266	255	245	113	224	234	202	189	177	167	246	246
113	334	374	368	362	355	283	341	348	327	320	313	305	299	292
134	327	368	362	355	348	275	334	341	320	313	305	298	292	284
149	320	362	355	348	341	268	327	334	313	305	298	290	284	277
167	313	355	348	341	334	257	320	327	305	298	290	283	277	269
177	305	348	341	334	327	247	313	320	298	290	283	275	269	259
189	298	341	334	327	320	236	305	313	290	283	275	268	259	249
202	290	334	327	320	313	225	298	305	283	275	268	257	249	246
224	275	320	313	305	298	203	283	290	268	257	247	236	246	246
234	268	313	305	298	290	191	275	283	257	247	236	225	246	246
245	257	305	298	290	283	179	268	275	247	236	225	215	246	246
266	236	290	283	275	268	152	247	257	225	215	203	191	246	246
274	225	283	275	268	257	137	236	247	215	203	191	179	246	246
0	341	380	374	368	362	290	348	355	334	327	320	313	307	299
341	0	189	177	167	149	202	113	134	113	134	149	167	246	246
380	189	0	113	134	149	266	177	167	201	213	223	234	246	254
374	177	113	0	113	134	255	167	149	189	201	213	223	246	246
368	167	134	113	0	113	245	149	134	177	189	201	213	246	246
362	149	149	134	113	0	234	134	113	167	177	189	202	246	246
290	202	266	255	245	234	0	213	224	189	177	167	149	246	246

348	113	177	167	149	134	213	0	113	134	149	167	177	246	246
355	134	167	149	134	113	224	113	0	149	167	177	189	246	246
334	113	201	189	177	167	189	134	149	0	113	134	149	246	246
327	134	213	201	189	177	177	149	167	113	0	113	134	246	246
320	149	223	213	201	189	167	167	177	134	113	0	113	246	246
313	167	234	223	213	202	149	177	189	149	134	113	0	246	246
307	246	246	246	246	246	246	246	246	246	246	246	246	0	113
299	246	254	246	246	246	246	246	246	246	246	246	246	113	0
207	344	383	377	371	365	294	351	358	337	330	323	316	310	303
207	333	373	367	361	354	281	340	347	326	319	311	304	298	290
204	321	363	356	349	342	269	328	335	314	306	299	291	285	278
204	299	342	335	328	321	238	306	314	291	284	276	269	261	250
204	289	333	326	319	311	223	296	304	281	274	266	255	247	236
218	279	323	316	309	301	209	286	294	271	262	252	241	232	222
233	269	314	306	299	291	204	276	284	259	248	238	227	218	207
261	241	294	286	279	271	204	252	262	230	220	209	204	187	175
273	227	284	276	269	259	204	238	248	216	205	204	204	172	160
290	204	266	255	245	234	204	213	223	204	204	204	204	160	160
310	204	237	227	216	205	204	204	204	204	204	204	204	160	160
329	204	209	204	204	204	204	204	204	204	204	204	204	160	169
247	255	304	296	289	281	204	266	274	245	234	223	213	203	191
297	191	257	246	236	225	177	203	214	178	177	177	177	187	187
177	305	348	341	334	327	246	312	320	298	290	283	275	269	259
194	295	338	331	324	317	232	303	310	288	280	273	264	256	245
224	275	320	312	305	298	203	283	290	268	257	246	236	227	216
238	264	310	303	295	288	187	273	280	254	243	232	222	213	202
252	250	300	293	285	278	177	261	270	239	229	218	207	198	187
335	177	199	187	177	177	192	177	177	177	177	177	177	187	187
307	177	243	232	222	211	177	187	199	177	177	177	177	187	187
316	177	229	218	207	195	177	177	182	177	177	177	177	187	187
326	177	214	203	191	178	177	177	177	177	177	177	177	187	187
345	177	182	177	177	177	207	177	177	177	177	177	177	187	194
354	177	177	177	177	177	222	177	177	177	177	177	187	198	209
363	177	177	177	177	177	236	177	177	177	179	192	204	213	224
177	315	357	350	343	336	261	322	329	308	300	293	285	279	272

135	**136**	**137**	**138**	**139**	**140**	**141**	**142**	**143**	**144**	**145**	**146**	**147**	**148**	**149**
189	**113**	**113**	**113**	**234**	**213**	**113**	**149**	**113**	**113**	**113**	**246**	**113**	**303**	**130**

149	150	151	152	153	154	155	156	157	158	159	160	161	162	163
368	368	370	370	370	370	370	370	370	370	370	370	370	384	384
368	368	370	370	370	370	370	370	370	370	370	370	370	384	384
368	368	370	370	370	370	370	370	370	370	370	370	370	384	384
368	368	370	370	370	370	370	370	370	370	370	370	370	384	384
382	372	362	344	344	344	344	344	344	344	344	344	344	359	359
376	366	355	344	344	344	344	344	344	344	344	344	344	359	359
370	359	348	329	329	329	329	329	329	329	329	329	329	345	345
364	352	341	329	329	329	329	329	329	329	329	329	329	345	345
357	345	334	329	329	329	329	329	329	329	329	329	329	345	345
350	341	344	344	344	344	344	344	344	344	344	344	344	359	359
343	331	329	329	329	329	329	329	329	329	329	329	329	345	345
341	341	344	344	344	344	344	344	344	344	344	344	344	359	359
329	326	329	329	329	329	329	329	329	329	329	329	329	345	345
341	341	344	344	344	344	344	344	344	344	344	344	344	359	359
326	326	329	329	329	329	329	329	329	329	329	329	329	345	345
341	341	344	344	344	344	344	344	344	344	344	344	344	359	359
326	326	329	329	329	329	329	329	329	329	329	329	329	345	345
341	341	344	344	344	344	344	344	344	344	344	344	344	359	359
326	326	329	329	329	329	329	329	329	329	329	329	329	345	345
341	341	344	344	344	344	344	344	344	344	344	344	344	359	359
341	341	344	344	344	344	344	344	344	344	344	344	344	359	359
326	326	329	329	329	329	329	329	329	329	329	329	329	345	345
341	341	344	344	344	344	344	344	344	344	344	344	344	359	359
326	326	329	329	329	329	329	329	329	329	329	329	329	345	345
341	341	344	344	344	344	344	344	344	344	344	344	344	359	359
326	326	329	329	329	329	329	329	329	329	329	329	329	345	345
341	341	344	344	344	344	344	344	344	344	344	344	344	359	359
326	326	329	329	329	329	329	329	329	329	329	329	329	345	345
341	341	344	344	344	344	344	344	344	344	344	344	344	359	359
326	326	329	329	329	329	329	329	329	329	329	329	329	345	345
326	326	329	329	329	329	329	329	329	329	329	329	329	345	345
341	341	344	344	344	344	344	344	344	344	344	344	344	359	359
382	372	362	341	331	322	312	293	283	276	276	276	303	294	347
376	366	355	334	324	315	305	289	289	289	289	289	295	307	340
370	359	348	327	317	307	298	278	276	276	276	276	288	294	333
368	357	345	324	315	305	295	289	289	289	289	289	289	307	330
364	352	341	320	310	300	290	276	276	276	276	276	280	294	326
359	347	335	314	304	294	284	263	249	224	215	215	274	240	320
357	345	334	312	303	293	283	276	276	276	276	276	276	294	319
350	338	327	305	295	285	276	276	276	276	276	276	276	294	311
348	337	325	303	293	283	276	276	276	276	276	276	276	294	309
343	331	320	298	288	278	268	261	261	261	261	261	261	279	304
338	326	314	292	282	272	261	261	261	261	261	261	261	279	298
336	324	312	290	280	270	261	261	261	261	261	261	261	279	296

329	317	305	283	273	261	261	261	261	261	261	261	261	279	289
328	316	304	281	271	261	261	261	261	261	261	261	261	279	287
322	310	298	275	264	261	261	261	261	261	261	261	261	279	281
317	305	293	270	261	261	261	261	261	261	261	261	261	279	279
315	303	290	276	276	276	276	276	276	276	276	276	276	294	294
308	295	289	289	289	289	289	289	289	289	289	289	289	307	307
307	294	289	289	289	289	289	289	289	289	289	289	289	307	307
300	288	276	276	276	276	276	276	276	276	276	276	276	294	294
296	287	289	289	289	289	289	289	289	289	289	289	289	307	307
293	280	276	276	276	276	276	276	276	276	276	276	276	294	294
287	287	289	289	289	289	289	289	289	289	289	289	289	307	307
278	274	276	276	276	276	276	276	276	276	276	276	276	294	294
287	287	289	289	289	289	289	289	289	289	289	289	289	307	307
274	274	276	276	276	276	276	276	276	276	276	276	276	294	294
287	287	289	289	289	289	289	289	289	289	289	289	289	307	307
274	274	276	276	276	276	276	276	276	276	276	276	276	294	294
287	287	289	289	289	289	289	289	289	289	289	289	289	307	307
274	274	276	276	276	276	276	276	276	276	276	276	276	294	294
326	326	329	329	329	329	329	329	329	329	329	329	329	345	345
287	287	289	289	289	289	289	289	289	289	289	289	289	307	307
326	326	329	329	329	329	329	329	329	329	329	329	329	345	345
274	274	276	276	276	276	276	276	276	276	276	276	276	294	294
287	287	289	289	289	289	289	289	289	289	289	289	289	307	307
274	274	276	276	276	276	276	276	276	276	276	278	276	294	294
258	258	261	261	261	261	261	261	261	261	261	283	261	279	279
258	258	261	261	261	261	261	261	261	261	265	285	261	279	279
258	258	261	261	261	261	261	261	261	261	273	293	261	279	279
258	258	261	261	261	261	261	261	261	261	274	294	261	279	279
258	258	261	261	261	261	261	261	261	261	280	300	261	279	279
258	258	261	261	261	261	261	261	261	264	285	305	261	279	279
258	258	261	261	261	261	261	261	261	268	288	308	261	279	279
258	258	261	261	261	261	261	261	261	275	295	315	261	282	279
258	258	261	261	261	261	261	261	265	283	303	322	261	289	279
388	377	367	347	337	328	319	299	289	271	245	216	309	264	352
373	363	351	330	321	311	301	281	271	248	239	239	291	263	336
364	352	341	320	310	300	290	270	257	232	203	202	280	227	326
354	342	330	309	299	289	279	255	241	216	202	202	269	227	315
343	331	320	298	288	278	268	261	261	261	261	261	261	279	304
333	321	309	286	276	266	266	266	266	266	266	266	266	284	293
322	310	298	275	264	250	249	249	249	249	249	249	249	270	281
311	299	286	262	248	234	220	202	202	202	202	202	205	227	270
300	288	275	249	249	249	249	249	249	249	249	249	249	270	270
289	276	266	266	266	266	266	266	266	266	266	266	266	284	284
271	255	249	249	249	249	249	249	249	249	249	249	249	270	270
257	239	222	202	202	202	202	202	202	202	204	233	202	227	227
241	223	211	211	211	211	211	211	211	211	220	249	211	236	236
236	236	239	239	239	239	239	239	239	239	239	265	239	263	263
211	211	215	215	215	215	215	215	215	224	252	277	215	240	240

208	208	211	211	211	211	211	211	215	240	268	288	211	249	236
208	208	211	211	211	211	211	217	231	256	279	299	211	265	236
208	208	211	211	211	211	211	233	247	270	290	310	218	277	236
388	377	367	347	337	328	319	299	289	271	245	216	309	264	352
354	342	330	309	299	289	279	255	241	239	239	239	269	263	315
343	331	320	298	288	278	268	239	225	211	211	211	254	236	304
322	310	298	275	264	250	236	211	211	211	211	211	222	236	281
311	299	286	262	248	239	239	239	239	239	239	239	239	263	270
300	288	275	246	232	218	215	215	215	215	215	215	215	240	255
289	276	262	239	239	239	239	239	239	239	239	239	239	263	263
271	255	238	211	211	211	211	211	211	211	211	217	211	236	236
257	239	222	211	211	211	211	211	211	211	211	233	211	236	236
225	211	215	215	215	215	215	215	215	215	236	265	215	240	240
236	236	239	239	239	239	239	239	239	239	252	277	239	263	263
208	208	211	211	211	211	211	211	215	240	268	288	211	249	236
208	208	211	211	211	211	211	217	231	256	279	299	211	265	236
208	208	211	211	211	211	211	233	247	270	290	310	218	277	236
205	185	167	160	160	160	160	185	202	227	256	279	171	236	187
154	154	160	160	174	189	206	234	249	272	292	312	220	278	187
348	336	324	303	293	283	273	246	232	207	175	136	261	197	309
323	311	299	276	266	252	238	209	193	167	126	126	223	152	283
316	304	291	269	255	241	227	197	180	149	83	139	213	136	275
286	274	259	227	213	197	180	143	126	113	162	194	167	132	236
268	250	232	199	182	169	146	71	120	159	192	222	128	171	209
234	216	197	161	138	129	129	150	171	198	227	256	130	207	172
149	133	129	156	174	189	206	234	249	272	292	312	220	278	141
333	333	336	336	336	336	336	336	336	336	336	336	336	352	352
207	207	204	204	215	229	243	270	280	298	318	336	258	304	189
286	274	259	227	213	204	204	204	204	204	204	204	204	177	236
207	207	204	204	204	207	222	250	265	283	303	322	236	289	177
207	207	204	204	204	204	211	240	254	275	295	315	225	282	177
207	207	204	204	204	204	204	229	243	268	288	308	215	274	177
207	207	204	204	204	204	204	218	233	257	280	300	204	266	177
207	207	204	204	204	204	204	207	222	247	273	293	204	256	177
207	207	204	204	204	204	204	204	211	236	265	285	204	245	177
207	207	204	204	204	204	204	204	204	225	254	278	204	234	177
229	211	204	204	204	204	204	204	204	204	233	261	204	213	177
239	222	204	204	204	204	204	204	204	204	222	250	204	202	177
250	232	214	204	204	204	204	204	204	204	211	240	204	189	189
270	254	236	204	204	204	204	204	204	204	204	218	204	177	213
278	264	246	214	204	204	204	204	204	204	204	207	204	177	223
207	207	204	204	204	218	233	261	273	290	310	329	247	297	177
344	333	321	299	289	279	269	241	227	204	204	204	255	191	305
383	373	363	342	333	323	314	294	284	266	237	209	304	257	348
377	367	356	335	326	316	306	286	276	255	227	204	296	246	341
371	361	349	328	319	309	299	279	269	245	216	204	289	236	334
365	354	342	321	311	301	291	271	259	234	205	204	281	225	327
294	281	269	238	223	209	204	204	204	204	204	204	204	177	246

351	340	328	306	296	286	276	252	238	213	204	204	266	203	312
358	347	335	314	304	294	284	262	248	223	204	204	274	214	320
337	326	314	291	281	271	259	230	216	204	204	204	245	178	298
330	319	306	284	274	262	248	220	205	204	204	204	234	177	290
323	311	299	276	266	252	238	209	204	204	204	204	223	177	283
316	304	291	269	255	241	227	204	204	204	204	204	213	177	275
310	298	285	261	247	232	218	187	172	160	160	160	203	187	269
303	290	278	250	236	222	207	175	160	160	160	169	191	187	259
0	130	155	194	209	224	238	266	277	294	314	333	252	300	183
130	0	130	174	189	205	220	248	263	282	301	321	234	288	165
155	130	0	149	170	185	202	231	245	269	289	309	216	275	137
194	174	149	0	126	143	167	198	213	238	266	286	181	247	129
209	189	170	126	0	126	143	181	198	224	252	277	167	232	137
224	205	185	143	126	0	126	167	181	209	238	266	143	218	159
238	220	202	167	143	126	0	143	167	194	224	252	126	203	175
266	248	231	198	181	167	143	0	126	162	194	224	126	172	207
277	263	245	213	198	181	167	126	0	139	177	209	143	152	222
294	282	269	238	224	209	194	162	139	0	143	181	177	129	247
314	301	289	266	252	238	224	194	177	143	0	143	209	132	273
333	321	309	286	277	266	252	224	209	181	143	0	238	172	293
252	234	216	181	167	143	126	126	143	177	209	238	0	187	191
300	288	275	247	232	218	203	172	152	129	132	172	187	0	256
183	165	137	129	137	159	175	207	222	247	273	293	191	256	0
200	179	158	129	129	137	159	191	207	232	261	283	175	241	126
229	211	191	152	132	129	129	159	175	203	232	261	137	213	167
243	225	207	172	152	132	129	137	159	187	218	247	129	198	181
257	240	222	187	172	152	132	129	137	172	203	232	129	181	198
339	327	315	293	283	273	261	232	218	191	158	129	247	181	299
310	298	285	261	247	232	218	187	172	137	129	152	203	126	269
320	308	295	273	261	247	232	203	187	158	129	132	218	143	279
329	318	305	283	273	261	247	218	203	175	137	129	232	167	289
348	336	325	303	293	283	273	247	232	207	175	137	261	198	309
357	346	334	313	303	293	283	261	247	222	191	158	273	213	319
366	355	343	322	313	303	293	273	261	236	207	175	283	227	328
169	141	129	137	159	175	191	222	236	261	283	303	207	269	126

150	**151**	**152**	**153**	**154**	**155**	**156**	**157**	**158**	**159**	**160**	**161**	**162**	**163**	**164**
130	**149**	**126**	**126**	**126**	**143**	**126**	**139**	**143**	**143**	**238**	**187**	**256**	**126**	**143**

164	165	166	167	168	169	170	171	172	173	174	175
384	384	384	384	384	384	384	384	384	384	384	384
384	384	384	384	384	384	384	384	384	384	384	384
384	384	384	384	384	384	384	384	384	384	384	384
384	384	384	384	384	384	384	384	384	384	384	384
359	359	359	359	359	359	359	359	359	359	359	359
359	359	359	359	359	359	359	359	359	359	359	359
345	345	345	345	345	345	345	345	345	345	345	345
345	345	345	345	345	345	345	345	345	345	345	345
345	345	345	345	345	345	345	345	345	345	345	345
359	359	359	359	359	359	359	359	359	359	359	359
345	345	345	345	345	345	345	345	345	345	345	345
359	359	359	359	359	359	359	359	359	359	359	359
345	345	345	345	345	345	345	345	345	345	345	345
359	359	359	359	359	359	359	359	359	359	359	359
345	345	345	345	345	345	345	345	345	345	345	345
359	359	359	359	359	359	359	359	359	359	359	359
345	345	345	345	345	345	345	345	345	345	345	345
359	359	359	359	359	359	359	359	359	359	359	359
345	345	345	345	345	345	345	345	345	345	345	345
359	359	359	359	359	359	359	359	359	359	359	359
359	359	359	359	359	359	359	359	359	359	359	359
345	345	345	345	345	345	345	345	345	345	345	345
359	359	359	359	359	359	359	359	359	359	359	359
345	345	345	345	345	345	345	345	345	345	345	345
359	359	359	359	359	359	359	359	359	359	359	359
345	345	345	345	345	345	345	345	345	345	345	345
359	359	359	359	359	359	359	359	359	359	359	359
345	345	345	345	345	345	345	345	345	345	345	345
359	359	359	359	359	359	359	359	359	359	359	359
345	345	345	345	345	345	345	345	345	345	349	345
345	345	345	345	345	345	345	345	345	347	356	345
359	359	359	359	359	359	359	359	359	359	363	359
337	319	309	299	294	294	294	294	294	294	294	356
330	311	307	307	307	307	307	307	307	307	307	349
323	304	294	294	294	294	294	294	294	294	294	342
321	307	307	307	307	307	307	307	307	307	307	340
316	296	294	294	294	294	294	294	294	294	294	335
310	290	280	270	240	240	240	240	240	240	240	329
309	294	294	294	294	294	294	294	294	294	294	328
301	294	294	294	294	294	294	294	294	294	294	321
299	294	294	294	294	294	294	294	294	294	294	319
294	279	279	279	279	279	279	279	279	279	279	314
288	279	279	279	279	279	279	279	279	279	279	308
286	279	279	279	279	279	279	279	279	279	279	306

279	279	279	279	279	279	279	279	279	279	279	299
279	279	279	279	279	279	279	279	279	279	279	297
279	279	279	279	279	279	279	279	279	279	279	291
279	279	279	279	279	279	279	279	279	279	279	286
294	294	294	294	294	294	294	294	294	294	294	294
307	307	307	307	307	307	307	307	307	307	307	307
307	307	307	307	307	307	307	307	307	307	307	307
294	294	294	294	294	294	294	294	294	294	294	294
307	307	307	307	307	307	307	307	307	307	307	307
294	294	294	294	294	294	294	294	294	294	294	294
307	307	307	307	307	307	307	307	307	307	307	307
294	294	294	294	294	294	294	294	294	294	294	294
307	307	307	307	307	307	307	307	307	307	307	307
294	294	294	294	294	294	294	294	294	294	294	294
307	307	307	307	307	307	307	307	307	307	307	307
294	294	294	294	294	294	294	294	294	294	294	294
307	307	307	307	307	307	307	307	307	307	307	307
294	294	294	294	294	294	294	294	294	294	294	294
345	345	345	345	345	345	345	345	345	345	345	345
307	307	307	307	307	307	307	307	307	307	307	307
345	345	345	345	345	345	345	345	345	345	345	345
294	294	294	294	294	294	294	294	294	297	307	294
307	307	307	307	307	307	307	307	307	307	308	307
294	294	294	294	294	294	294	294	294	304	314	294
279	279	279	279	289	279	279	279	299	309	319	279
279	279	279	279	292	279	279	282	302	312	321	279
279	279	279	279	299	279	279	289	309	319	328	279
279	279	279	279	300	279	280	290	310	320	329	279
279	279	279	279	307	279	287	297	316	326	335	279
279	279	279	279	311	281	291	301	321	330	339	279
279	279	279	279	314	284	294	304	324	333	342	279
279	279	279	279	321	292	302	312	331	340	349	279
279	279	279	279	328	299	309	319	338	347	356	279
343	324	315	305	236	250	236	236	236	236	236	362
327	307	298	288	263	263	263	263	263	263	263	345
316	296	286	276	227	227	227	227	227	227	227	335
305	285	275	264	227	227	227	227	227	227	227	324
294	279	279	279	279	279	279	279	279	279	279	314
284	284	284	284	284	284	284	284	284	284	284	303
271	270	270	270	270	270	270	270	270	270	270	291
257	229	227	227	227	227	227	227	227	227	227	280
270	270	270	270	270	270	270	270	270	270	270	270
284	284	284	284	284	284	284	284	284	284	284	284
270	270	270	270	270	270	270	270	270	270	270	270
227	227	227	227	241	227	227	227	256	269	279	227
236	236	236	236	257	236	236	243	270	280	290	236
263	263	263	263	272	263	263	263	282	292	302	263
240	240	240	240	283	247	261	273	293	303	313	240

236	236	236	236	294	263	274	284	304	314	324	236
236	236	236	236	305	275	285	295	315	325	334	236
236	236	236	236	316	287	297	307	326	335	345	236
343	324	315	305	207	250	236	221	191	187	187	362
305	285	275	264	263	263	263	263	263	263	263	324
294	274	262	248	236	236	236	236	236	236	236	314
271	245	236	236	236	236	236	236	236	236	236	291
263	263	263	263	263	263	263	263	263	263	263	280
241	240	240	240	240	240	240	240	240	240	240	269
263	263	263	263	263	263	263	263	263	263	263	263
236	236	236	236	236	236	236	236	240	254	268	236
236	236	236	236	241	236	236	236	256	269	279	236
240	240	240	240	272	240	245	259	282	292	302	240
263	263	263	263	283	263	263	273	293	303	313	263
236	236	236	236	294	263	274	284	304	314	324	236
236	236	236	236	305	275	285	295	315	325	334	236
236	236	236	236	316	287	297	307	326	335	345	236
187	187	187	187	285	250	265	275	295	305	315	187
187	196	211	225	318	288	298	308	327	336	346	187
299	279	269	255	129	180	167	143	129	129	144	319
273	246	232	218	137	132	129	129	159	176	192	293
264	236	222	207	153	129	129	133	172	187	204	285
222	191	175	158	204	153	172	187	218	233	247	250
193	161	138	129	231	185	202	217	245	259	272	223
152	90	107	137	265	222	236	250	275	285	295	187
165	196	211	225	318	288	298	308	327	336	346	120
352	352	352	352	352	352	352	352	352	354	363	352
206	234	249	263	342	314	324	333	352	361	369	177
222	191	177	177	204	177	177	187	218	233	247	250
181	213	227	241	328	299	309	319	338	347	356	177
177	202	217	231	321	292	302	312	331	340	349	177
177	189	206	220	314	284	294	304	324	333	342	177
177	177	194	209	307	277	287	297	316	326	335	177
177	177	181	198	299	269	279	289	309	319	328	177
177	177	177	185	292	259	272	282	302	312	321	177
177	177	177	177	284	249	263	274	294	304	314	177
177	177	177	177	269	227	241	256	279	289	299	180
177	177	177	177	259	217	231	245	272	282	292	193
177	177	177	177	249	206	220	234	263	274	284	205
197	177	177	177	227	181	198	213	241	256	269	227
209	177	177	177	217	177	185	202	231	245	259	238
194	224	238	252	335	307	316	326	345	354	363	177
295	275	264	250	177	177	177	177	177	177	177	315
338	320	310	300	199	243	229	214	182	177	177	357
331	312	303	293	187	232	218	203	177	177	177	350
324	305	295	285	177	222	207	191	177	177	177	343
317	298	288	278	177	211	195	178	177	177	177	336
232	203	187	177	192	177	177	177	207	222	236	261

303	283	273	261	177	187	177	177	177	177	177	322
310	290	280	270	177	199	182	177	177	177	177	329
288	268	254	239	177	177	177	177	177	177	177	308
280	257	243	229	177	177	177	177	177	177	179	300
273	246	232	218	177	177	177	177	177	177	192	293
264	236	222	207	177	177	177	177	177	187	204	285
256	227	213	198	187	187	187	187	187	198	213	279
245	216	202	187	187	187	187	187	194	209	224	272
200	229	243	257	339	310	320	329	348	357	366	169
179	211	225	240	327	298	308	318	336	346	355	141
158	191	207	222	315	285	295	305	325	334	343	129
129	152	172	187	293	261	273	283	303	313	322	137
129	132	152	172	283	247	261	273	293	303	313	159
137	129	132	152	273	232	247	261	283	293	303	175
159	129	129	132	261	218	232	247	273	283	293	191
191	159	137	129	232	187	203	218	247	261	273	222
207	175	159	137	218	172	187	203	232	247	261	236
232	203	187	172	191	137	158	175	207	222	236	261
261	232	218	203	158	129	129	137	175	191	207	283
283	261	247	232	129	152	132	129	137	158	175	303
175	137	129	129	247	203	218	232	261	273	283	207
241	213	198	181	181	126	143	167	198	213	227	269
126	167	181	198	299	269	279	289	309	319	328	126
0	143	167	181	289	256	269	279	299	309	319	143
143	0	126	143	269	227	241	256	279	289	299	181
167	126	0	126	256	213	227	241	269	279	289	198
181	143	126	0	241	198	213	227	256	269	279	213
289	269	256	241	0	167	143	126	126	143	167	309
256	227	213	198	167	0	126	143	181	198	213	279
269	241	227	213	143	126	0	126	167	181	198	289
279	256	241	227	126	143	126	0	143	167	181	299
299	279	269	256	126	181	167	143	0	126	143	319
309	289	279	269	143	198	181	167	126	0	126	328
319	299	289	279	167	213	198	181	143	126	0	337
143	181	198	213	309	279	289	299	319	328	337	0

165	**166**	**167**	**168**	**169**	**170**	**171**	**172**	**173**	**174**	**175**	**1**
126	**126**	**241**	**167**	**126**	**126**	**143**	**126**	**126**	**337**	**384**	

SOLUCIÓN DEL TSP8 USANDO INDICE Y SOLVER

8

	1	2	3	4	5	6	7	8
1	0	3	4	8	8	9	9	10
2	3	0	4	4	10	9	9	7
3	4	4	0	6	6	5	5	10
4	8	4	6	0	12	11	8	3
5	8	10	6	12	0	4	10	14
6	9	9	5	11	4	0	6	10
7	9	9	5	8	10	6	0	4
8	10	7	10	3	14	10	4	0

VECTOR

8 7 6 5 3 1 2 4 8

DISTANCIAS

4 6 4 6 4 3 4 3

SOLUCIÓN

34

SOLUCIÓN DEL TSP7 USANDO INDICE Y SOLVER

7

	1	2	3	4	5	6	7
1	0	3	4	8	8	9	9
2	3	0	4	4	10	9	9
3	4	4	0	6	6	5	5
4	8	4	6	0	12	11	8
5	8	10	6	12	0	4	10
6	9	9	5	11	4	0	6
7	9	9	5	8	10	6	0

VECTOR

7 4 2 1 3 5 6 7

DISTANCIAS

8 4 3 4 6 4 6

SOLUCIÓN

35

SOLUCIÓN DEL TSP6 USANDO INDICE Y SOLVER

6

	1	2	3	4	5	6
1	0	702	454	842	2396	1196
2	702	0	324	1093	2136	764
3	454	324	0	1137	2180	798
4	842	1093	1137	0	1616	1857
5	2396	2136	2180	1616	0	2900
6	1196	764	798	1857	2900	0

VECTOR

1	**3**	**6**	**2**	**5**	**4**	**1**

DISTANCIAS

454	**798**	**764**	**2136**	**1616**	**842**

SOLUCIÓN

6610

SOLUCIÓN DEL SWISS42 USANDO INDICE Y SOLVER

42

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	15	30	23	32	55	33	37	92	114	92	110	96
2	15	0	34	23	27	40	19	32	93	117	88	100	87
3	30	34	0	11	18	57	36	65	62	84	64	89	76
4	23	23	11	0	11	48	26	54	70	94	69	89	75
5	32	27	18	11	0	40	20	58	67	92	61	78	65
6	55	40	57	48	40	0	23	55	96	123	78	75	62
7	33	19	36	26	20	23	0	45	85	111	75	82	69
8	37	32	65	54	58	55	45	0	124	149	118	126	113
9	92	93	62	70	67	96	85	124	0	28	29	68	63
10	114	117	84	94	92	123	111	149	28	0	54	91	88
11	92	88	64	69	61	78	75	118	29	54	0	39	34
12	110	100	89	89	78	75	82	126	68	91	39	0	14
13	96	87	76	75	65	62	69	113	63	88	34	14	0
14	90	75	93	84	76	36	60	80	122	150	99	80	72
15	74	63	95	84	83	56	63	42	148	174	134	129	117
16	76	67	100	89	89	66	70	42	155	181	142	139	128
17	82	71	104	92	91	63	71	49	156	182	141	135	124
18	67	69	98	89	95	95	85	40	159	181	157	167	153
19	72	62	57	54	43	37	44	87	67	95	44	39	26
20	78	63	88	78	72	34	52	60	129	157	110	98	88
21	82	96	99	99	110	137	115	94	148	159	161	187	174
22	159	164	130	141	141	174	161	195	78	50	103	136	136
23	122	132	100	111	116	156	136	158	80	65	109	148	142
24	131	131	101	109	105	129	122	163	39	27	52	81	82
25	206	212	179	190	190	224	210	242	129	102	154	186	187
26	112	106	86	89	81	90	91	135	46	65	22	28	32
27	57	44	51	44	34	15	25	65	82	110	63	61	48
28	28	33	4	11	19	59	37	63	65	87	68	92	79
29	43	51	18	29	35	75	54	79	55	73	66	97	85
30	70	77	43	54	57	96	78	106	40	50	61	98	89
31	65	75	45	56	63	103	81	101	61	68	81	117	106
32	66	72	95	89	97	105	90	50	157	176	158	173	159
33	37	52	45	47	58	91	68	66	97	112	107	134	121
34	103	118	115	118	129	158	136	118	159	166	175	204	191
35	84	99	93	96	107	139	116	104	135	142	151	181	168
36	125	132	152	147	156	164	150	109	212	229	216	232	219
37	129	132	159	151	158	156	147	103	221	241	219	229	216
38	72	67	100	90	92	78	76	36	159	184	150	153	140
39	126	139	112	122	129	169	148	160	110	99	137	176	168
40	141	148	114	126	127	163	147	178	72	46	100	137	134
41	183	186	153	163	161	191	180	218	95	69	115	143	145
42	124	122	94	101	95	115	111	153	35	38	37	62	64

VECTOR													
25	29	3	4	1	2	5	6	27	13	26	11	9	23
DISTANCIAS													
163	18	11	23	15	27	40	15	48	32	22	29	80	36
SOLUCIÓN													
1864													

14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
90	74	76	82	67	72	78	82	159	122	131	206	112	57	28
75	63	67	71	69	62	63	96	164	132	131	212	106	44	33
93	95	100	104	98	57	88	99	130	100	101	179	86	51	4
84	84	89	92	89	54	78	99	141	111	109	190	89	44	11
76	83	89	91	95	43	72	110	141	116	105	190	81	34	19
36	56	66	63	95	37	34	137	174	156	129	224	90	15	59
60	63	70	71	85	44	52	115	161	136	122	210	91	25	37
80	42	42	49	40	87	60	94	195	158	163	242	135	65	63
122	148	155	156	159	67	129	148	78	80	39	129	46	82	65
150	174	181	182	181	95	157	159	50	65	27	102	65	110	87
99	134	142	141	157	44	110	161	103	109	52	154	22	63	68
80	129	139	135	167	39	98	187	136	148	81	186	28	61	92
72	117	128	124	153	26	88	174	136	142	82	187	32	48	79
0	59	71	63	116	56	25	170	201	189	151	252	104	44	95
59	0	11	8	63	93	35	135	223	195	184	273	146	71	95
71	11	0	11	54	103	46	130	230	198	192	279	155	80	99
63	8	11	0	65	100	39	140	232	203	192	281	153	78	103
116	63	54	65	0	127	92	83	224	180	199	269	175	106	95
56	93	103	100	127	0	67	153	145	139	96	196	53	23	60
25	35	46	39	92	67	0	152	207	188	162	258	119	48	89
170	135	130	140	83	153	152	0	188	128	184	222	183	139	95
201	223	230	232	224	145	207	188	0	65	57	51	109	160	132
189	195	198	203	180	139	188	128	65	0	91	94	126	145	100
151	184	192	192	199	96	162	184	57	91	0	106	53	115	104
252	273	279	281	269	196	258	222	51	94	106	0	158	211	180
104	146	155	153	175	53	119	183	109	126	53	158	0	75	89
44	71	80	78	106	23	48	139	160	145	115	211	75	0	53
95	95	99	103	95	60	89	95	132	100	104	180	89	53	0
111	113	117	121	109	70	107	95	116	82	94	163	88	68	18
130	138	143	147	135	81	129	110	90	60	74	136	83	86	44
138	138	141	146	125	95	134	91	102	57	94	145	103	95	45
130	81	74	85	21	134	108	62	217	167	196	259	178	114	92
127	107	107	115	80	101	114	54	148	99	134	190	129	90	42
192	159	155	164	107	172	176	24	188	126	192	218	197	160	112
174	146	143	152	100	149	159	23	168	106	168	200	173	139	89
186	132	122	133	71	194	163	81	264	208	251	302	236	173	149
172	113	102	112	63	190	147	110	281	230	260	323	238	168	156
90	32	22	33	33	115	66	113	231	194	197	278	166	92	99
205	200	202	208	173	160	200	108	100	36	126	120	156	162	111
193	209	215	218	205	138	197	164	26	39	64	65	111	150	116
214	243	250	251	249	159	224	217	30	94	64	49	115	176	155
135	171	179	178	191	80	147	184	75	103	19	124	34	101	97

39	**30**	**31**	**33**	**35**	**34**	**21**	**28**	**7**	**19**	**12**	**14**	**20**	**15**	**17**
79	**21**	**46**	**49**	**24**	**24**	**95**	**37**	**44**	**39**	**80**	**25**	**35**	**8**	**11**

29	30	31	32	33	34	35	36	37	38	39	40	41	42
43	70	65	66	37	103	84	125	129	72	126	141	183	124
51	77	75	72	52	118	99	132	132	67	139	148	186	122
18	43	45	95	45	115	93	152	159	100	112	114	153	94
29	54	56	89	47	118	96	147	151	90	122	126	163	101
35	57	63	97	58	129	107	156	158	92	129	127	161	95
75	96	103	105	91	158	139	164	156	78	169	163	191	115
54	78	81	90	68	136	116	150	147	76	148	147	180	111
79	106	101	50	66	118	104	109	103	36	160	178	218	153
55	40	61	157	97	159	135	212	221	159	110	72	95	35
73	50	68	176	112	166	142	229	241	184	99	46	69	38
66	61	81	158	107	175	151	216	219	150	137	100	115	37
97	98	117	173	134	204	181	232	229	153	176	137	143	62
85	89	106	159	121	191	168	219	216	140	168	134	145	64
111	130	138	130	127	192	174	186	172	90	205	193	214	135
113	138	138	81	107	159	146	132	113	32	200	209	243	171
117	143	141	74	107	155	143	122	102	22	202	215	250	179
121	147	146	85	115	164	152	133	112	33	208	218	251	178
109	135	125	21	80	107	100	71	63	33	173	205	249	191
70	81	95	134	101	172	149	194	190	115	160	138	159	80
107	129	134	108	114	176	159	163	147	66	200	197	224	147
95	110	91	62	54	24	23	81	110	113	108	164	217	184
116	90	102	217	148	188	168	264	281	231	100	26	30	75
82	60	57	167	99	126	106	208	230	194	36	39	94	103
94	74	94	196	134	192	168	251	260	197	126	64	64	19
163	136	145	259	190	218	200	302	323	278	120	65	49	124
88	83	103	178	129	197	173	236	238	166	156	111	115	34
68	86	95	114	90	160	139	173	168	92	162	150	176	101
18	44	45	92	42	112	89	149	156	99	111	116	155	97
0	27	27	103	42	109	85	157	168	115	94	98	140	90
27	0	21	128	62	119	96	179	192	142	79	72	115	74
27	21	0	115	46	98	75	163	179	136	67	81	129	95
103	128	115	0	69	86	81	60	65	54	158	195	243	190
42	62	46	69	0	71	49	117	133	98	95	127	175	132
109	119	98	86	71	0	24	94	127	137	100	163	218	194
85	96	75	81	49	24	0	104	133	127	85	143	197	170
157	179	163	60	117	94	104	0	39	100	190	241	292	246
168	192	179	65	133	127	133	39	0	81	216	259	307	253
115	142	136	54	98	137	127	100	81	0	193	214	253	187
94	79	67	158	95	100	85	190	216	193	0	74	129	137
98	72	81	195	127	163	143	241	259	214	74	0	55	80
140	115	129	243	175	218	197	292	307	253	129	55	0	81
90	74	95	190	132	194	170	246	253	187	137	80	81	0

16	**8**	**18**	**32**	**36**	**37**	**38**	**24**	**42**	**10**	**40**	**22**	**41**	**25**
42	**40**	**21**	**60**	**39**	**81**	**197**	**19**	**38**	**46**	**26**	**30**	**49**	