
Trabajo Práctico 3

Compilado: 20 de octubre de 2023

Fecha de entrega: 15 de noviembre de 2023 (antes de las 23:59)

Fecha de reentrega: 4 de diciembre de 2023 (antes de las 23:59)

Este trabajo práctico consta de cuatro ejercicios que pueden ser resueltos mediante la aplicación de las técnicas algorítmicas vistas en clase. Para cada uno de ellos se deberá diseñar e implementar un algoritmo que lo resuelva, y es necesario que dicha implementación supere una prueba de eficiencia (tanto temporal como espacial) en un juez online.

En todos los ejercicios la instancia se leerá de la entrada estándar y la solución se imprimirá en la salida estándar. Junto a los enunciados se incluyen algunos casos de test que sirven como ejemplo tanto del formato de salida como del problema en cuestión. Está **terminantemente prohibida** la copia de código, ya sea de otro grupo o de internet. La detección de plagio puede resultar en la desaprobación directa de la materia o en otras medidas más graves.

El trabajo práctico se debe realizar en grupos de a 3. Para aprobarlo es necesario aprobar cada ejercicio en forma individual, ya sea en la primera entrega o en el recuperatorio. No es necesario reentregar aquellos ejercicios que sean aprobados en la primera entrega. Asimismo, para aprobar cada ejercicio es necesario que la implementación del algoritmo propuesto pase la prueba del juez online y que este sea entregado a los docentes para su revisión.

Formato de entrega La entrega se realizará por medio del Campus, llenando un cuestionario que se habilitará a tal efecto. El código se tomará de las entregas que el grupo haga en el sistema SPOJ (se pedirá el usuario correspondiente al grupo en el cuestionario).

Ejercicio 1: Salvataje

Esta vez Pablo Lamponne debe salvar a su perro Betún, que se encuentra atrapado al final de una fortaleza. La fortaleza consta de una serie de N salones rectangulares paralelos divididos por murallas, que debe ir saltando de a una, hasta llegar al último salón donde se encuentra su amigo cuadrúpedo.

A Pablo le lleva un minuto escalar cada muralla, pero conoce que algunos salones poseen pasadizos subterráneos que puede recorrer en dos minutos y de esta manera adelantar salones sin tener que escalar sus murallas. Sin embargo, los túneles son muy frágiles y Pablo no quiere arriesgarse a que se ocasione un derrumbe, por lo que planea utilizar a lo sumo 3 pasadizos en su operativo de rescate. La fortaleza está diseñada para evitar todo tipo de escape, por lo que no es posible escalar una muralla para pasar de un salón al anterior ni acceder a uno de los túneles desde su salida. ¿Cuál es el mínimo tiempo posible que le tomará a Pablo llegar hasta Betún?

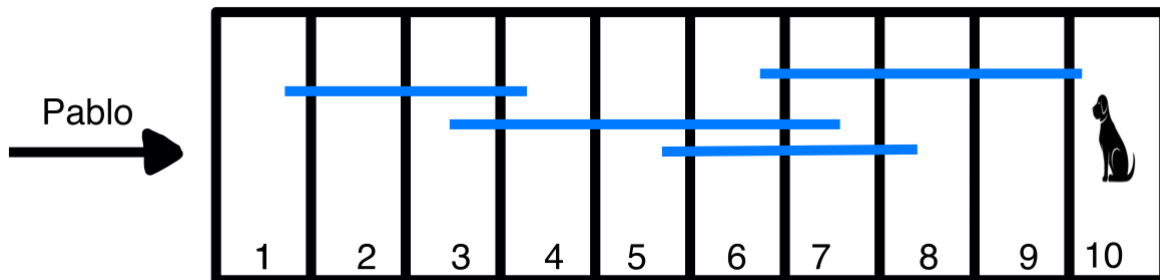


Figura 1: Una fortaleza de 10 salones y 4 pasadizos secretos en azul ($N = 10$ y $M = 4$).

Descripción de una instancia

La entrada contiene múltiples casos de test. La primera línea consiste de un número c indicando su cantidad.

La primera línea de cada caso contiene dos enteros N y M , indicando respectivamente la cantidad de salones de la fortaleza y la cantidad de pasadizos secretos. Luego vienen M líneas con 2 números que indican el comienzo y el final de cada pasadizo. Vale que $1 \leq N \leq 10^6$ y $1 \leq M \leq 10^7$.

Descripción de la salida

Para cada caso de test se debe imprimir una línea indicando el mínimo tiempo (en minutos) que le toma a Pablo llegar hasta Betún.

Ejemplo de entrada y salida Se presenta un ejemplo de entrada y su correspondiente salida.

Entrada de ejemplo	Salida esperada de ejemplo
3	3
4 2	6
1 4	7
2 4	10
6 1	
3 5	
10 4	
1 4	
3 7	
5 8	
6 10	
16 4	
1 5	
5 8	
8 13	
13 16	

En el primer caso de test el tiempo mínimo es de 3 minutos dado que el pasadizo secreto va del salón 1 hasta el final directamente (entonces a Pablo le toma 1 minuto entrar en el salón 1 y luego 2 minutos atravesar el pasadizo secreto). Pablo nunca toma el segundo pasadizo.

En el segundo caso de test, Pablo se tarda 3 minutos en llegar al salón 3, luego (tome o no el pasadizo) le toma 2 minutos más llegar al salón 5 y un minuto más llegar al último salón. En total son 6 minutos.

En el tercer caso de test (correspondiente a la figura 1), se puede ver que le conviene tomar el pasadizo 1 y luego el pasadizo 4, logrando encontrar a Betún en 7 minutos.

Finalmente, en el cuarto caso de test, Pablo potencialmente podría tomar los 4 pasadizos, pero debido a que puede tomar a lo sumo 3, le conviene aprovechar el que lo lleva del salón 1 al 5 y el que lo lleva del salón 8 al 13. Ya sea que elija tomar el que lo lleva del 5 al 8 o el del 13 al 16, el tiempo total terminará siendo de 10 minutos.

Ejercicio 2: Distancias

Una anécdota que Pablo Lamponne recuerda de su época haciendo mantenimiento de redes es aquella vez en que tuvo que deducir las distancias de conexión de la red de un edificio a partir de sus latencias. En aquel entonces, Pablo recibió un llamado donde le querían consultar sobre la posibilidad de instalar nuevos enlaces en la red de un edificio, para lo cual el respondió que necesitaba conocer las distancias de conexión entre todos los pares de computadoras en la red. En esta red, todas las computadoras estaban conectadas, directa o indirectamente, unas con otras, por medio de enlaces bidireccionales. La *distancia de conexión* entre dos computadoras u y v se define como la mínima cantidad de enlaces por los que debe pasar un paquete enviado desde u por la red para llegar hasta v .

Sin embargo, por un malentendido, la información que terminó recibiendo Pablo es la de la *latencia* entre todo par de computadoras, que es la mínima cantidad de tiempo, medida en una cantidad de milisegundos entera y mayor que 0, que toma a un paquete enviado desde u para llegar hasta v (potencialmente pasando por otras computadoras, el tiempo durante el paso por una computadora intermedia se considera despreciable). La latencia entre u y v puede no ser proporcional a la distancia de conexión entre u y v , porque los enlaces pueden tener distinta latencia, pero aún así Pablo se convenció de que no necesitaba pedir que le enviaran la información de las distancias de conexión. Esto se debe a que a Pablo le mencionaron que la red era *eficiente*, es decir, que tenía la cantidad de enlaces mínima que permitía que se dieran las latencias comunicadas. Con ese dato, Pablo supo que podía obtener las distancias de conexión entre todas las computadoras sabiendo las latencias entre todas las computadoras, o incluso darse cuenta si las latencias comunicadas eran incompatibles con una red posible. ¿Podremos hacerlo nosotros también?

Descripción de una instancia

La entrada contiene múltiples casos de test. La primera línea consiste de un número c indicando su cantidad.

La primera línea de cada caso contiene un entero N ($1 \leq N \leq 100$) que es la cantidad de computadoras en la red. Luego le siguen $N - 1$ líneas. La i -ésima línea tiene i valores que corresponden con la latencia l_{ij} , ($1 \leq l_{ij} \leq 10^4$) entre la computadora i y la computadora j , con $j < i$.

Descripción de la salida

Para cada caso de test se debe imprimir “IMPOSIBLE”, si los valores provistos no son consistentes con las distancias de una red, o “POSIBLE”, si sí lo son. En este caso, se debe continuar con N líneas, donde la i -ésima línea debe tener los valores de las distancias de conexión entre la computadora i y cada computadora j , para todo $0 \leq j \leq N$.

Ejemplo de entrada y salida Se presenta un ejemplo de entrada y su correspondiente salida.

Entrada de ejemplo	Salida esperada de ejemplo
4	POSIBLE
3	0 1 2
100	1 0 1
200 100	2 1 0
3	IMPOSIBLE
100	POSIBLE
300 100	0 1 1
3	1 0 1
100	1 1 0
50 100	POSIBLE
4	0 1 2 2
100	1 0 1 1
300 200	2 1 0 1
200 100 200	2 1 1 0

En estos ejemplos, podemos ver que el primero es posible con dos enlaces, por lo que la distancia de conexión entre la computadora 1 y la 2 es 1, entre la 2 y la 3 es 1, y entre la 3 y la 1 es 2.

La segunda tabla de latencias no es posible, ya que si la latencia entre la computadora 1 y la 2 es 100 y entre la 2 y la 3 es 100, no es posible que la latencia entre la 1 y la 3 sea 300.

En el tercer caso, la distancia de conexión entre todas las computadoras es 1, pues no hay manera de alcanzar la latencia informada si no hay una conexión directa entre cada par de computadoras.

Finalmente, en el cuarto ejemplo, vemos una red de 4 computadoras donde la latencia entre la computadora 1 y la 4 se consigue recorriendo los enlaces 1 a 2 y 2 a 4.

La Figura 2 muestra las estructuras para los tres casos posibles.

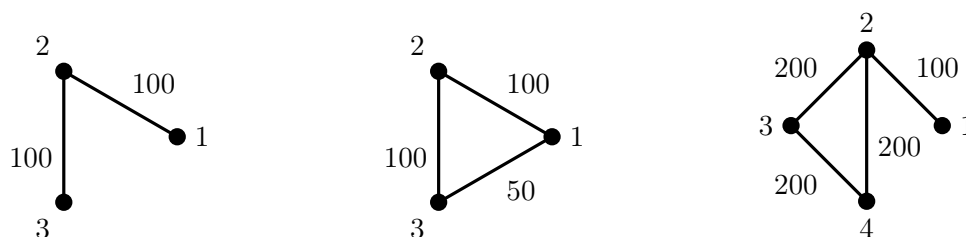


Figura 2: Estructuras para las redes del primer, tercer y cuarto ejemplo (el segundo no es posible)

Ejercicio 3: Torres

En una cómoda tarde de primavera, Lamponne y Santos se encontraban jugando no tan apaciblemente al ajedrez. Tras su tercer derrota, Lamponne arrojó el tablero de mármol por los aires, logrando como resultando que tras la caída varios casilleros del mismo se resquebrajaran.

Santos, en vez de enojarse, observó con curiosidad que ahora, en este tablero con casilleros “tapados”, era posible colocar más de una torre por fila sin que estas se amenazen, si se interpreta que dos torres se amenazan mutuamente si y solamente si se encuentran en la misma fila (o columna) y no hay un casillero roto entre ellos. Interesado en esta cualidad, se propuso encontrar la máxima cantidad de torres que se pueden colocar en este tablero sin que se amenazen mutuamente.

Lamponne, que sigue herido por las tres derrotas, nos pide ayuda para calcular este número antes que Santos.

Descripción de una instancia

La entrada contiene múltiples casos de test. La primera línea consiste de un número c indicando su cantidad.

La primera línea de cada caso contiene un entero N ($1 \leq N \leq 100$), indicando la cantidad de filas y columnas del tablero. Luego siguen N líneas con N enteros cada una, describiendo el mismo. Si el j -ésimo carácter de la i -ésima fila es un 1, entonces ese casillero (el de la fila i y columna j) está roto, mientras que caso contrario, si el entero es un 0, entonces está intacto.

Descripción de la salida

Para cada caso de test se debe imprimir una línea indicando la máxima cantidad de torres que pueden colocarse en este tablero resquebrajado siguiendo las reglas que dispuso Santos.

Ejemplo de entrada y salida Se presenta un ejemplo de entrada y su correspondiente salida.

Entrada de ejemplo	Salida esperada de ejemplo
3	2
2	4
0 0	3
0 1	
3	
0 0 0	
0 1 0	
0 0 0	
3	
1 0 0	
1 1 0	
0 0 1	

En el primer caso de test es posible colocar dos torres: una en la segunda posición de la primera fila, y la otra en la primera posición de la segunda fila.

Las formas de colocar las torres en el segundo y el tercer caso de test se pueden ver a continuación:

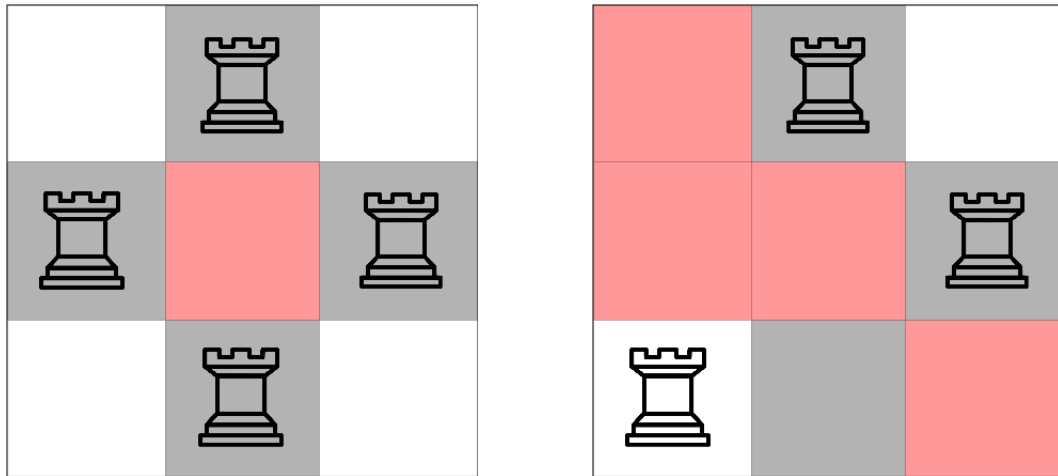


Figura 3: Formas de colocar la máxima cantidad de torres posibles para el segundo y tercer caso de test.

Ejercicio 4: Envíos

20 años más tarde, Lamponne y su equipo son contactados para una última misión. Esta requiere una enorme cantidad de recursos y herramientas, por lo que contactan a todos sus anteriores compañeros para poder prepararse. Aparte, nos piden ayuda para administrar a esta enorme cantidad de personas.

Si bien no nos informan de qué se trata este último encargo, nos dicen que necesitan mover muchísimos artilugios desde su taller hasta la Casa Rosada. Su idea es que cada uno de los x compañeros contactados lleve algunos artefactos a través de las calles, pero debido a la discreción que requiere toda esta operación establecieron una cota c_e a la cantidad de herramientas que pueden transportarse a través de la cada calle e . Por último, como es natural esperar de ellos, quieren que este trabajo se organice de forma **justa**: desean que cada persona lleve una misma cantidad de artefactos desde el taller hasta la Casa Rosada.

Dados estos requerimientos, debemos informarles cuál es la máxima cantidad de instrumentos que van a poder llevar desde su taller hasta la Casa Rosada, considerando que cada colaborador puede hacer un único viaje.

Descripción de una instancia

La entrada contiene múltiples casos de test. La primera línea consiste de un número c indicando su cantidad.

La primera línea de cada caso de test contiene tres enteros N, M y x . El entero N ($1 \leq N \leq 100$) denota la cantidad de esquinas de la ciudad, mientras que M ($1 \leq M \leq 1000$) denota la cantidad de calles. El taller del equipo se encuentra en la esquina 1, mientras que la Casa Rosada se ubica en la esquina N . El tercer entero x ($1 \leq x \leq 10^6$) señala la cantidad de compañeros que se pudieron contactar.

Luego siguen M líneas describiendo las calles. Cada línea tiene tres enteros v, w y c : v indica la esquina de origen de la calle, w la de destino (es decir, las calles tienen **dirección**), y c ($1 \leq c \leq 10^6$) es la cantidad máxima de herramientas que pueden transportarse sobre la calle vw .

Descripción de la salida

Para cada caso de test se debe imprimir un entero indicando la máxima cantidad de herramientas que pueden transportarse del taller a la Casa Rosada respetando los límites de transporte por calle, asegurando que la organización del proceso sea **justa** y considerando que cada persona puede viajar del taller a la Casa Rosada una única vez.

Ejemplo de entrada y salida Se presenta un ejemplo de entrada y su correspondiente salida.

Entrada de ejemplo	Salida esperada de ejemplo
3	6
3 3 3	8
1 3 3	0
1 2 4	
2 3 5	
4 4 2	
1 2 15	
2 4 7	
2 3 4	
3 4 7	
4 4 5	
1 2 2	
1 3 7	
2 4 5	
3 4 2	

En el primer caso de test, es posible hacer que cada una de las tres personas lleve dos herramientas: la primera persona puede tomar el camino directo (la calle $1 \rightarrow 3$), mientras que las otras dos pueden usar el camino $1 \rightarrow 2 \rightarrow 3$). Es imposible hacer que lleven tres herramientas cada una y que no se superen los límites de transporte por calle. Luego, la respuesta es 6.

En el segundo caso de test, es posible hacer que cada persona lleve 4 herramientas: la primera sigue el camino $1 \rightarrow 2 \rightarrow 4$, mientras que la segunda hace $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$. Por lo tanto, la respuesta es 8.

En el tercer caso de test ninguna es posible asignarle una herramienta a cada persona sin que se supere el límite de transporte por calle. Luego, la respuesta es 0.