30
Los amigos de mis amigos son mis amigos

En esta ciudad viven N personas, y sabemos que algunas de ellas son amigas entre sí. De acuerdo con el refrán que dice “Los amigos de mis amigos son mis amigos”, sabemos que si A y B son amigos y B y C son amigos, entonces también son amigos A y C.

Tu misión consiste en contar las personas en el grupo de amigos más grande.

Entrada

La entrada consta de varios casos de prueba. La primera línea contiene un número que indica el número de casos de prueba que vendrán a continuación.

La primera línea de cada caso contiene dos números: el número N de personas que viven en la ciudad ($1 \leq N \leq 20.000$) y el número M de pares de personas que se conoce que son amigas ($0 \leq M \leq 200.000$). A continuación aparecen M líneas cada una con dos enteros A y B ($1 \leq A,B \leq N$; $A \neq B$) que indican que A y B son amigos.

Salida

Para cada caso de prueba se escribirá una línea con el número de personas en el grupo de amigos más grande.

Entrada de ejemplo

```
2
3 2
1 2
2 3
10 10
1 2
3 1
3 4
5 4
3 5
4 6
5 2
7 10
9 10
8 9
```

Salida de ejemplo

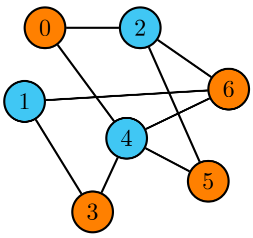
```
3
6
```

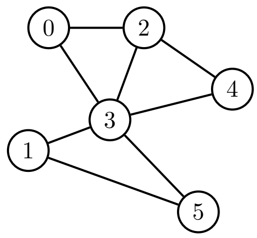
Autor: Alberto Verdejo.

31
Grafo bipartito

Un grafo no dirigido es *bipartito* si sus vértices pueden repartirse en dos conjuntos disjuntos de tal forma que todas las aristas tengan un extremo en cada uno de esos conjuntos.

Dicho de otra forma, un grafo es bipartito si sus vértices pueden colorearse utilizando dos colores de tal forma que no exista ninguna arista que conecte dos vértices del mismo color. De los dos grafos no dirigidos siguientes, el de la izquierda es bipartito, pero el de la derecha no lo es.





Entrada

La entrada está compuesta por diversos casos de prueba. Para cada caso, la primera línea contiene el número de vértices del grafo, V (entre 1 y 100), y la segunda el número de aristas, A. A continuación aparecen A líneas, cada una con dos enteros que representan los extremos de cada una de las aristas (valores entre 0 y $V-1$). Los grafos no contienen aristas de un vértice a sí mismo ni más de una arista que conecte un mismo par de vértices.

Salida

Para cada caso de prueba se escribirá en una línea independiente la palabra `SI` si el grafo es bipartito y `NO` en caso contrario.

Entrada de ejemplo

```
7
9
0 2
0 4
1 6
1 3
2 6
2 5
4 6
4 5
4 3
6
8
0 2
0 3
2 3
2 4
4 3
3 1
3 5
1 5
```

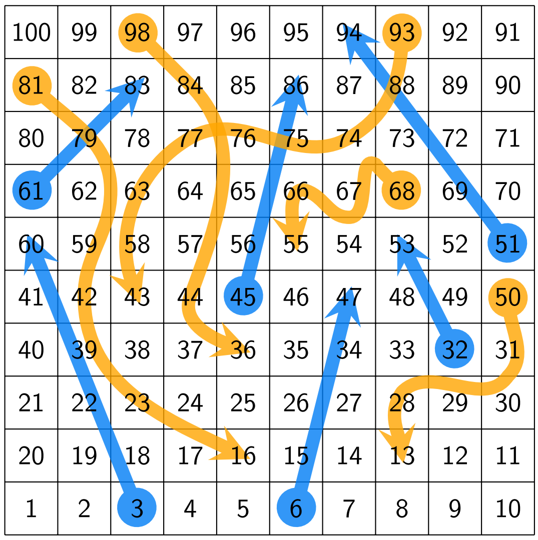
Salida de ejemplo

```
SI
NO
```

Autor: Alberto Verdejo.

32
Haciendo trampas en Serpientes y Escaleras

Serpientes y Escaleras es un juego clásico, originario de la India, donde ya se jugaba en el siglo XVI. El tablero está formado por una cuadrícula de $N \times N$ casillas numeradas de forma consecutiva desde 1 hasta N^2, comenzando por la esquina inferior izquierda y continuando fila por fila de abajo arriba, alternando en cada fila el ir hacia la izquierda o hacia la derecha, como aparece en el dibujo. Algunos pares de casillas, siempre en filas diferentes, pueden estar conectados mediante *serpientes* (que bajan, naranjas en el dibujo) o *escaleras* (que suben, azules en el dibujo). Cada casilla puede ser extremo de como mucho una serpiente o una escalera. La primera y la última casilla nunca son extremos de una serpiente o escalera.



A *Serpientes y Escaleras* pueden jugar cualquier número de jugadores correspondiéndole a cada uno una ficha. Todas las fichas comienzan en la casilla número 1. Los jugadores van alternándose para mover su ficha. Para ello, tiran un dado con K caras numeradas desde 1 hasta K, y avanzan su ficha siguiendo la numeración del tablero tantas casillas como indique el dado. Si la ficha termina en el extremo superior de una serpiente, se deslizará hasta su extremo inferior. En cambio, si termina en el extremo inferior de una escalera, ascenderá hasta su extremo superior. Gana la partida el jugador que antes alcance la última casilla.

El juego así planteado no requiere ninguna destreza. Pero supongamos que has trucado el dado y tienes el poder de elegir la cara que saldrá cada vez que lo tiras. ¿Sabes cuántos movimientos tendrías que hacer para ganar la partida si comienzas moviendo tú?

Entrada

La entrada consta de una serie de casos de prueba. En la primera línea de cada caso aparecen cuatro números: la dimensión N del tablero, el número K de caras del dado ($K \leq N$), el número S de serpientes y el número E de escaleras. Las siguientes $S + E$ líneas contienen cada una dos números, indicando la casilla inicial y la casilla final de una serpiente (las S primeras) o una escalera (las E siguientes). Tanto N como K son números entre 1 y 100.

La entrada termina con `0 0 0 0`.

Salida

Para cada caso de prueba se escribirá el menor número de movimientos necesarios para ganar la partida. Está garantizado que la casilla final es alcanzable desde la inicial en todos los casos.

Entrada de ejemplo

```
10 6 5 6
50 13
68 55
81 16
93 43
98 36
3 60
6 47
32 53
45 86
51 94
61 83
0 0 0 0
```

Salida de ejemplo

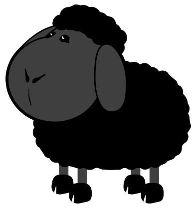
```
3
```

Notas

El caso de prueba del ejemplo se corresponde con la figura.

Autor: Alberto Verdejo.

33
Ovejas negras



Parece ser que en cierta ocasión estaban de viaje por Escocia un abogado, un físico y un matemático. Por la ventanilla del tren en el que viajaban vieron un campo con ovejas negras. Ninguno de los tres había visto ovejas negras nunca, por lo que se estableció un curioso diálogo entre ellos:

— ¡Vaya! ¡En Escocia las ovejas son negras! —dijo el abogado.

— Querrás decir que en Escocia *algunas* ovejas son negras... —corrigió el físico.

— Bueno, —no pudo evitar decir el matemático— con lo que hemos visto lo único que podemos decir es que en Escocia algunas ovejas son negras... *¡por un lado!*

Para comprobar si el abogado tenía razón o no, se han tomado fotografías a todas las ovejas de Escocia, y ahora hay que analizarlas para ver si aparecen ovejas blancas (al menos por un lado).

Entrada

La entrada estará compuesta por distintos casos de prueba, cada uno siendo una instantánea de una o más ovejas escocesas. Cada foto comienza con una línea con dos números indicando el ancho y el alto (en píxeles) de la imagen. A continuación viene la imagen en blanco y negro en donde el carácter '`.`' representa el color blanco y '`X`' el negro.

Se puede asumir que:

- El fondo de la imagen es siempre blanco.
- Todas las ovejas tienen la silueta negra. Las ovejas blancas tienen todo blanco en su interior.
- Las ovejas nunca se solapan (es decir, en las fotos las ovejas nunca se tocan).
- Ninguna oveja entra en contacto con los bordes de la foto (es decir, en todas las fotos la primera y última fila y la primera y última columna son '`.`').
- En la foto solo aparecen ovejas.

Salida

Para cada caso de prueba se escribirá el número de ovejas blancas que aparecen en la foto.

Entrada de ejemplo

```
16 8
................
..........XXX...
.....XXXXXXXX...
....XXXXXXX.....
.....XXXXX......
.....X...X......
.....X...X......
................
22 8
......................
.......XXX..XXX.......
..XXXXXXXX..XXXXXXXX..
.X.....X......XXXXXXX.
..XXXXX........XXXXX..
..X...X........X...X..
..X...X........X...X..
......................
```

Salida de ejemplo

```
0
1
```

Autores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

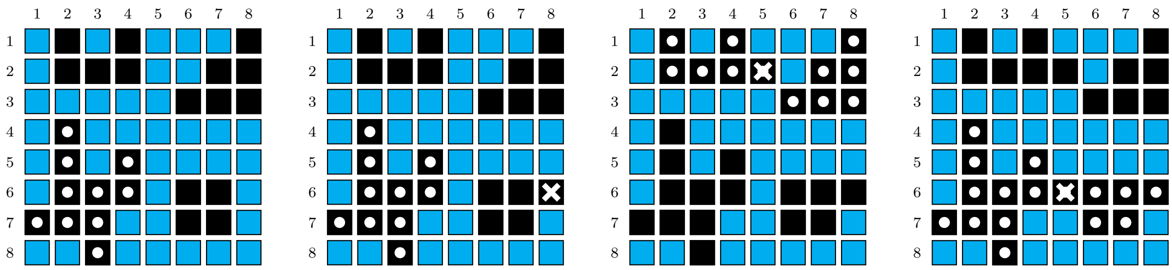
34

Petroleros hundidos

Desgraciadamente, las asociaciones ecologistas se ven obligadas a hacer frente, periódicamente, a las llamadas "*mareas negras*". En ellas, grandes cantidades de crudo son vertidas al mar, dejando enormes superficies de agua con petróleo flotando a la deriva.

Para estimar los daños medioambientales, se realizan fotografías de las zonas afectadas utilizando satélites geoestacionarios. La superficie del mar queda dividida en una rejilla de celdas, cada una marcada como zona contaminada o como zona limpia (al menos de momento). Las imágenes obtenidas son también usadas para organizar los trabajos de limpieza, para los que no es importante la superficie total contaminada, sino la superficie *contigua* más grande (*mancha* más grande). Dada una fotografía del satélite, dos celdas contaminadas de petróleo (negras) se consideran pertenecientes a la misma mancha si se puede llegar de una a otra atravesando solo celdas contaminadas realizando desplazamientos en cualquiera de las 8 direcciones (horizontal, vertical, y dos diagonales).

Cuando el petrolero se hunde, muchas veces sigue derramando crudo que, al emerger, aumenta la zona contaminada y puede hacer que cambie cual es la mayor mancha. Por ejemplo, en el siguiente esquema se muestra la primera imagen que se tomó, y la evolución horaria al ir subiendo más crudo desde las profundidades del mar. En cada una, se marca con puntos blancos las celdas de la mancha más grande (con su tamaño en el pie de la imagen), y con una cruz la última celda que ha pasado a estar contaminada.



10 10 12 16

Entrada

La entrada estará compuesta por diversos casos de prueba. Para cada caso, la primera línea contendrá el número F de filas y el número C de columnas de la rejilla (números entre 1 y 1.000). A continuación aparecerán F líneas, cada una con C caracteres. El espacio en blanco representa una celda azul (mar) y el carácter `#` representa una celda contaminada de negro petróleo. En la siguiente línea aparecerá un número no negativo N (no mayor de 100.000) indicando el número de imágenes adicionales tomadas (en cada una aparece una nueva celda de petróleo), seguido de N líneas cada una con dos enteros que indicarán la fila (entre 1 y F) y columna (entre 1 y C) donde aparecerá esa celda contaminada.

Salida

Para cada caso de prueba se escribirá una línea con el tamaño de la mancha de petróleo más grande inicialmente, seguido de los tamaños tras añadir cada una de las nuevas celdas contaminadas, separados por espacios. Las superficies se miden en número de celdas.

Entrada de ejemplo

```
8 8
 # #   #
 ###  ##
     ###
 #
 # #
 ### ##
###  ##
  #
3
6 8
2 5
6 5
```

Salida de ejemplo

```
10 10 12 16
```

Autores: Alberto Verdejo y Pedro Pablo Gómez Martín.

35
La ardilla viajera

La leyenda popular dice que en el libro *Geografía* escrito en el siglo I A.C., el geógrafo griego Estrabón dijo de la península ibérica que era tan frondosa que una ardilla podía cruzarla de sur a norte saltando de árbol en árbol sin tocar el suelo.

Parece ser que, en realidad, el bueno de Estrabón nunca aseguró tal cosa en su libro y, de hecho, hoy en día se cree que ni siquiera en aquellos tiempos la hazaña de la ardilla pudo ser cierta.

No obstante dejemos volar la imaginación por un momento y pensemos que en algún instante del pasado el número de árboles en el territorio de la península permitiera la aventura. Dado que hoy día no es posible realizarla, está claro que en algún momento del pasado se cortó el árbol que provocó la separación de la región del norte y la del sur en lo que a saltos de rama en rama se refiere.

Para simplificar un poco el problema, asumamos que el territorio es una región cuadrada de tamaño $N \times M$ en la que se sitúan árboles (que consideraremos de grosor 0) en posiciones (x, y). La tarea de la ardilla consiste en ir desde el árbol situado en la posición $(0, 0)$ hasta el árbol en la posición (N, M) (en ambas posiciones hay siempre un árbol). La ardilla es capaz de saltar de un árbol a otro si la distancia entre ambos no supera K unidades.

Los datos que tenemos del territorio son las posiciones de todos los árboles al principio de los tiempos y el orden en el que fueron cortados y debemos determinar la posición del árbol que, al ser cortado, hizo que la ardilla no pudiera atravesar la península de parte a parte sin pisar el suelo.

Entrada

Cada caso de prueba se compone de varias líneas. La primera de ellas tiene los números N, M, K y n ($1 \leq N, M \leq 1.000$; $1 \leq K \leq 10$; $1 \leq n \leq 100.000$), donde los tres primeros tienen el significado descrito más arriba y el último indica el número de árboles en el territorio (sin contar los situados en la esquina origen y destino de la ardilla que siempre están presentes).

A continuación aparece una línea por cada árbol con dos enteros x, y con la posición de cada uno. El orden en el que se dan es el mismo en el que después fueron cortados. Se garantiza que todas las posiciones están dentro del territorio y que no hay dos árboles en la misma ubicación.

Salida

Por cada caso de prueba se escribirá una única línea con la posición dentro del territorio del árbol que, cuando se cortó, provocó que las dos esquinas del territorio no fueran alcanzables para la ardilla.

Si la ardilla nunco pudo atravesar la península de parte a parte, se escribirá `NUNCA SE PUDO`.

Entrada de ejemplo

```
3 3 2 4
1 1
2 2
2 0
3 1
3 3 2 2
3 0
0 3
```

Salida de ejemplo

```
2 0
NUNCA SE PUDO
```

Autor: Marco Antonio Gómez Martín.

36
Camiones de reparto

Somos una empresa de transporte y hemos decidido renovar parte de nuestra flota de camiones de reparto. A nosotros nos conviene que los camiones sean anchos, porque así se puede repartir y colocar mejor la mercancia. Pero claro, hay ciudades con calles muy estrechas, por donde no todos los camiones pueden pasar.

Tenemos mapas actualizados de las ciudades donde trabajamos, donde hemos señalado para cada calle cuál es la anchura máxima que puede tener un camión para poder transitar por ella.

¿Nos ayudas a decidir si un camión de una anchura determinada puede circular por una ciudad para llegar desde un punto concreto a otro?

Entrada

La entrada está formada por una serie de casos de prueba. En cada caso, primero se describe una ciudad. La primera línea contiene el número V de intersecciones de la ciudad (numeradas de 1 a V) y la segunda el número E de calles entre intersecciones. A continuación aparecen E líneas, cada una con tres números: las intersecciones que une esa calle, y la anchura máxima que puede tener un camión que transite por ella. Todas las calles son de doble sentido.

Tras la descripción de la ciudad, aparece un número K de consultas, seguido de K líneas, cada una con tres números: dos intersecciones distintas, el origen y el destino, y la anchura de un camión, del que estamos interesados en saber si podría viajar desde el origen hasta el destino.

Todos los casos cumplen que $2 \leq V \leq 10.000$, $0 \leq E \leq 100.000$ y $1 \leq K \leq 10$. Todas las anchuras son números entre 1 y 1.000.000.

Salida

Para cada caso de prueba se escribirán K líneas, una por consulta. La respuesta a una consulta será `SI` si un camión de la anchura correspondiente podría recorrer un camino que le llevara del origen al destino, y `NO` en caso contrario.

Entrada de ejemplo

```
5
5
1 2 10
1 3 30
2 4 20
3 4 15
4 5 12
3
1 5 8
1 4 12
2 5 15
```

Salida de ejemplo

```
SI
SI
NO
```

Autor: Alberto Verdejo.

37
Recorriendo el archipiélago en bicicleta

Al archipiélago *Milislotes* han llegado las bicicletas. Sus habitantes están estusiasmados y no se bajan de la bici en todo el día, yendo de un lado para otro sin descanso. Pero las islas son tan pequeñas que terminan todos mareados de dar vueltas sin parar, por lo que se están planteando crear una red de puentes rectos que les permita ir en bici de cualquier isla a cualquier otra.

Han pedido presupuesto al arquitecto, que ha confeccionado una lista con todos los puentes que podrían construirse entre islas del archipiélago y cuánto costaría construir cada uno de esos puentes. ¿Podrías ayudarles a decidir qué puentes construir de tal forma que se pueda ir en bici desde cualquier isla a cualquier otra y el coste total de la obra sea lo mínimo posible?

Entrada

La entrada está formada por varios casos de prueba. Cada uno ocupa varias líneas: en la primera aparece el número I de islas en el archipiélago (entre 1 y 1.000); en la siguiente aparece el número P de puentes presupuestados (entre 0 y 10.000); y a continuación aparece una línea por cada uno de estos puentes con tres enteros, las islas que une (numeradas entre 1 e I) y el coste de construir ese puente (un valor entre 1 y 100.000). Los puentes siempre van de una isla a otra distinta, son transitables en bici en ambos sentidos, y no se ha presupuestado más de un puente entre un mismo par de islas.

Salida

Para cada caso de prueba se escribirá, en una línea, el coste mínimo de construir los puentes necesarios para unir todas las islas por bici. Si no hubiera suficientes puentes presupuestados para lograrlo, se escribirá `No hay puentes suficientes`.

Entrada de ejemplo

```
4
5
1 2 5
1 3 10
2 4 7
1 4 8
3 4 2
4
3
1 2 3
2 4 5
4 1 8
```

Salida de ejemplo

```
14
No hay puentes suficientes
```

Autor: Alberto Verdejo.

38

Navegando sin teclado

Lo de ayer fue un drama. Con los años que llevas criticando a todos los que abusan del ratón en lugar de utilizar las teclas rápidas, te tuvo que pasar a ti. Maldito teclado, que se tuvo que romper. Y encima en día festivo, imposible comprar otro. Todo el día utilizando el ordenador a través del lento e incómodo ratón. Menos mal que lo único que hiciste fue navegar por internet y pudiste ir de una página a otra utilizando los enlaces entre ellas.

Una persona normal con el orgullo herido compraría un teclado de reserva. Pero tú no. Con un flamante teclado nuevo vas a hacer un programa que te ayude a decidir la manera más rápida de navegar si te vuelves a ver en la misma situación.

Para eso has recopilado todas las páginas web que visitas y has hecho una lista de los enlaces que tiene cada una de ellas. Teniendo en cuenta la página de inicio que tienes en el navegador quieres saber cuál es la forma más rápida de llegar a una página determinada utilizando únicamente esos enlaces.

Como estás obsesionado por la velocidad, el programa tendrá en cuenta que el tiempo de carga de cada página puede ser distinto y que una vez cargada, encontrar y pinchar en cada enlace depende de lo lejos que esté del principio de la página y de lo oculto que esté. Venga, ¡manos a la obra!

Entrada

La entrada estará compuesta por distintos casos de prueba terminados por una línea con un 0.

Cada caso de prueba representa un escenario de navegación distinto, que incluye el conjunto de páginas a visitar, sus tiempos de carga y los enlaces que hay entre ellos.

La primera línea de cada caso contiene el número N de páginas manejadas ($1 < N \leq 1.000$). La página de inicio del navegador es la 1 y la página que queremos visitar la N.

A continuación viene una línea con N números que contiene el tiempo de carga (en milisegundos) de cada una de las páginas.

Aparece después una línea con un número M indicando el número de enlaces entre páginas que se han identificado, tras lo que aparecen M líneas cada una describiendo un enlace con la página en la que está, la página a la que va y el tiempo (en milisegundos) que se tarda desde que la página carga hasta que el usuario encuentra el enlace y lo pulsa.

Se garantiza que todos los tiempos serán menores a 10 segundos y que en caso de que una página tenga más de un enlace a la misma página, únicamente se proporciona aquel que puede utilizarse antes.

Salida

Para cada caso de prueba se escribirá el número mínimo de milisegundos que se tardará en poder navegar desde la página 1 hasta la página N contando el tiempo de carga de todas las páginas (incluídas la 1 y la N) y el tiempo de localización y pulsación de cada enlace.

Si no se puede alcanzar la página, se escribirá `IMPOSIBLE`.

Entrada de ejemplo

```
4
10 5 15 8
5
1 2 10
1 3 20
1 4 100
2 4 20
3 4 20
4
10 5 15 8
4
1 2 10
1 3 20
2 3 40
4 2 10
0
```

Salida de ejemplo

```
53
IMPOSIBLE
```

Autores: Marco Antonio Gómez Martín y Alberto Verdejo.

39

Repartiendo paquetes

La comarca de *Dispersión de Arriba* está formada por multitud de bonitas casas repartidas por laderas y colinas. Las casas están conectadas por carreteras, carreterillas, caminos o senderos, dependiendo de las zonas y de la importancia de sus dueños. Las características de estas conexiones son muy variadas, desde anchas carreteras prácticamente planas a empinados senderos en ocasiones embarrados y muy poco transitables, sobre todo cuesta arriba.

La oficina de la empresa de transportes que atiende a la comarca está situada en una de estas casas. Cada día, un único repartidor debe entregar los paquetes recibidos, y para ello cuenta con una moto cuyo compartimento de carga es tan pequeño que solamente puede llevar un paquete cada vez. Con estas restricciones, la rutina del sufrido repartidor consiste en tomar uno de los paquetes, llevarlo hasta la casa del destinatario, y volver a la oficina a por el siguiente paquete. Debido a las condiciones de las vías, hay ocasiones en las que interesa, o es incluso inevitable, que el camino de vuelta siga una ruta distinta al de ida.

Conociendo el esfuerzo que supone para el repartidor viajar por las distintas vías en cada sentido transitable, y las casas donde debe entregar paquetes un día concreto, ¿puedes ayudarle a decidir la mejor forma (la de menor esfuerzo total) de repartir los paquetes? El repartidor tiene total libertad para elegir en qué orden reparte los paquetes y qué rutas sigue, tanto para ir como para volver.

Entrada

La entrada está compuesta por diversos casos de prueba, ocupando cada uno de ellos varias líneas.

La primera línea contiene el número N de casas en la comarca (entre 1 y 10.000) y la segunda el número C de conexiones directas entre casas (entre 1 y 100.000). A continuación, aparecen C líneas cada una con tres enteros, *origen destino esfuerzo*, que indican que se puede ir directamente desde la casa *origen* hasta la casa *destino* (las casas están numeradas desde 1 hasta N) con el consiguiente *esfuerzo* (un valor entre 1 y 10.000). Ten en cuenta que si una conexión es transitable en ambos sentidos aparecerá dos veces, con el origen y el destino intercambiados y con un esfuerzo posiblemente distinto para cada sentido (al fin y al cabo, no es lo mismo subir que bajar).

Tras la descripción de la comarca, aparece una línea con dos enteros: O es el número de la casa donde se encuentra la oficina (el repartidor debe comenzar y terminar su jornada laboral en esta casa) y P (entre 1 y N) es el número de paquetes a repartir. A continuacion, aparece otra línea con los P números de las casas donde viven los destinatarios.

Salida

Para cada caso de prueba el programa debe escribir una línea con el esfuerzo mínimo necesario para repartir todos los paquetes de ese día. Se garantiza que este valor nunca será mayor que 10^9. Si para un día el reparto es imposible porque no existen suficientes conexiones para construir las rutas necesarias, el programa escribirá `Imposible`.

Entrada de ejemplo

```
4
5
1 2 5
2 3 2
3 1 8
1 4 2
4 1 3
1 3
2 3 4
4
3
1 3 2
3 1 3
3 4 5
1 2
2 3
```

Salida de ejemplo

```
35
Imposible
```

Autor: Alberto Verdejo.