10. Árboles libres

Se dice que un grafo no dirigido es un *árbol libre* si es acíclico y conexo (o dicho de otra manera, todo par de vértices está conectado por exactamente un camino).

De los dos grafos siguientes, solamente el de la izquierda es árbol libre.

El problema consiste en decidir si un grafo no dirigido es árbol libre o no.

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 10.000), y la segunda el número de aristas, A (entre 0 y 100.000). 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 `SI` si el grafo es árbol libre y `NO` en caso contrario.

Entrada de ejemplo

```
6
5
0 5
0 2
2 1
2 3
4 3
6
5
0 1
1 2
2 3
3 0
4 5
```

Salida de ejemplo

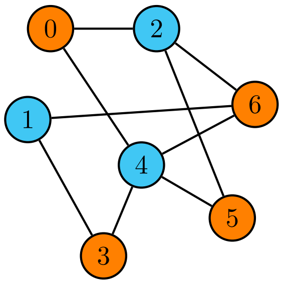
```
SI
NO
```

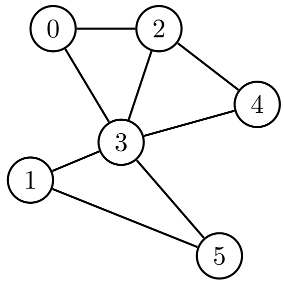
Autor: Alberto Verdejo.

11. 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 siguientes, el de la izquierda es bipartito, pero el de la derecha no lo es.





El problema consiste en determinar si un grafo no dirigido dado es bipartito o no.

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 10.000), y la segunda el número de aristas, A (entre 0 y 100.000). 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

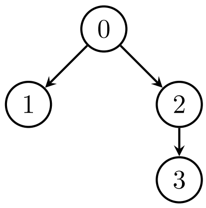
```
SI
NO
```

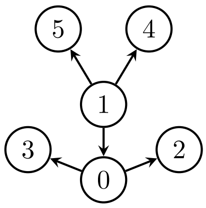
Autor: Alberto Verdejo.

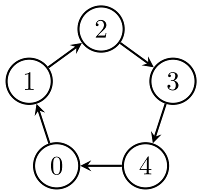
12. Arborescencias

Un grafo dirigido es una *arborescencia* si existe un vértice, llamado *raíz*, desde el que se puede alcanzar cualquier otro vértice a través de un camino único.

De los siguientes grafos, el de la izquierda es arborescencia con raíz en el vértice 0; el del centro también es arborescencia con raíz en el vértice 1; y el grafo de la derecha no es arborescencia.







El problema consiste en, dado un grafo dirigido, determinar si es arborescencia o no, y en caso de serlo, indicar qué vértice es la raíz.

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 10.000), y la segunda el número de aristas dirigidas, A (entre 0 y 100.000). A continuación aparecen A líneas, cada una con dos enteros que representan el origen y el destino de cada una de las aristas (valores entre 0 y $V-1$). Los grafos no contienen aristas de un vértice a sí mismo ni aristas repetidas.

Salida

Para cada caso de prueba se escribirá `SI` seguido del vértice raíz, si el grafo es arborencencia, y se escribirá `NO` en caso contrario.

Entrada de ejemplo

```
4
3
0 1
0 2
2 3
6
5
0 2
0 3
1 0
1 4
1 5
5
5
0 1
1 2
2 3
3 4
4 0
```

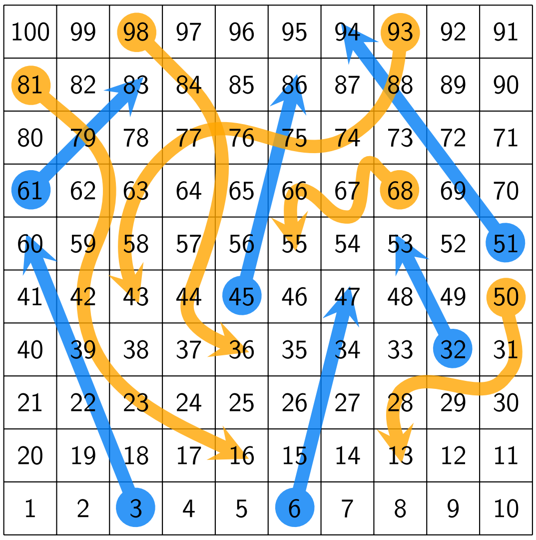
Salida de ejemplo

```
SI 0
SI 1
NO
```

Autor: Alberto Verdejo.

13. 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

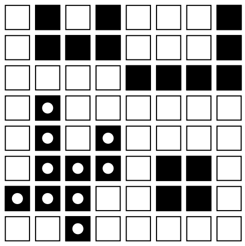
El caso de prueba del ejemplo se corresponde con la figura.

Autor: Alberto Verdejo.

14. Detección de manchas negras

Dado un *bitmap* de píxeles blancos y negros, queremos saber el tamaño (número de píxeles) de la *mancha* negra más grande. Dos píxeles negros pertenecen a la misma mancha si se puede pasar de uno a otro atravesando solamente píxeles negros y moviéndonos píxel a píxel solamente en horizontal o vertical.

Por ejemplo, en el siguiente dibujo (donde los píxeles se han representado mediante cuadrados) la mancha más grande (marcada con puntos blancos) tiene 10 píxeles.



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 del bitmap (números entre 1 y 1.000). A continuación aparecerán F líneas, cada una con C caracteres. El espacio en blanco representa un píxel blanco y el carácter `#` representa un píxel negro.

Salida

Para cada caso de prueba se escribirá en una línea independiente el tamaño de la mancha más grande.

Entrada de ejemplo

```
8 8
 # #   #
 ###   #
    ####
 #
 # #
 ### ##
###  ##
  #
4 10
###   ####
#     ####
##
##########
```

Salida de ejemplo

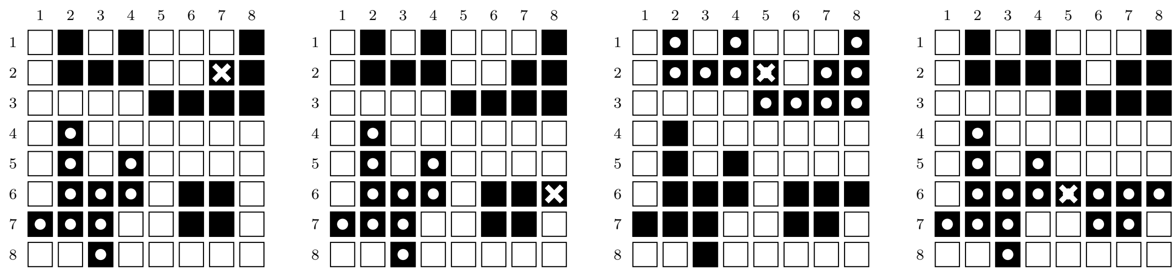
```
10
16
```

Autor: Alberto Verdejo.

15. Detección de manchas negras crecientes

De nuevo estamos interesados en el tamaño (número de píxeles) de la *mancha* negra más grande en un *bitmap* de píxeles blancos y negros. Dos píxeles negros pertenecen a la misma mancha si se puede pasar de uno a otro atravesando solamente píxeles negros y moviéndonos píxel a píxel solamente en horizontal o vertical.

En esta ocasión irán apareciendo nuevos píxeles negros, que harán que ciertas manchas vayan creciendo. Por ejemplo, en el siguiente dibujo (donde los píxeles se han representado mediante cuadrados), aparece debajo de cada bitmap el tamaño de la mancha más grande (marcada con puntos blancos). En cada caso el último píxel pintado de negro aparece señalado con una cruz.



10 10 13 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 del bitmap (números entre 1 y 1.000). A continuación aparecerán F líneas, cada una con C carácteres. El espacio en blanco representa un píxel blanco y el carácter # representa un píxel negro. En la siguiente línea aparecerá un número no negativo N (no mayor de 100.000) indicando el número de píxeles que se volverán negros, 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 ese píxel negro.

Salida

Para cada caso de prueba se escribirán, en una línea y separados por espacios, los tamaños de la mancha más grande después de añadir cada uno de los nuevos puntos negros.

Entrada de ejemplo

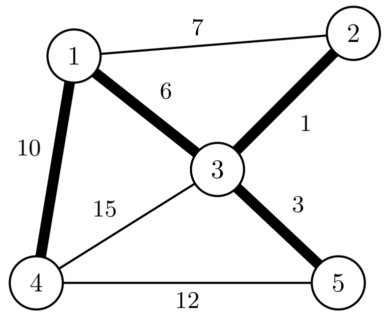
```
8 8
 # #   #
 ###   #
    ####
 #
 # #
 ### ##
###  ##
  #
4
2 7
6 8
2 5
6 5
```

Salida de ejemplo

```
10 10 13 16
```

Autor: Alberto Verdejo.

16. Pavimentar Barro City



Los residentes de Barro City son demasiado tacaños para pavimentar las calles de la ciudad; después de todo, a nadie le gusta pagar impuestos. Sin embargo, tras varios meses de lluvias intensas empiezan a estar cansados de enfangarse los pies cada vez que salen a la calle.

Debido a su gran tacañería, en vez de pavimentar todas las calles de la ciudad, quieren pavimentar solamente las suficientes para poder ir de una intersección a otra cualquiera de la ciudad siguiendo una ruta pavimentada y, además, quieren gastarse tan poco dinero como sea posible en la realización de esta obra. Y es que a los residentes de Barro City no les importa caminar mucho, si ello les permite ahorrar algún dinero. El alcalde tiene interés en saber cuál es el mínimo presupuesto que tiene que reservar de las arcas públicas para pavimentar la ciudad.

Por ejemplo, en una ciudad como la del dibujo con 5 intersecciones y 7 calles, donde el número que aparece al lado de cada calle representa lo que costaría pavimentarla, convendría pavimentar las calles que aparecen más gruesas.

Entrada

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

En la primera línea aparece el número N (entre 1 y 10.000) de intersecciones en la ciudad, y en la segunda línea aparece el número C (entre 0 y 100.000) de calles (entre intersecciones). A continuación, aparece una línea por cada calle con tres enteros, que indican los números de las intersecciones que une la calle (números entre 1 y N) y lo que costaría pavimentarla (un valor entre 1 y 100.000). Nunca hay una calle que conecte una intersección consigo misma, ni dos calles que conecten el mismo par de intersecciones.

Salida

Para cada caso de prueba se escribirá una línea con el coste mínimo necesario para pavimentar la ciudad de tal forma que se pueda viajar de cualquier intersección a cualquier otra por calles pavimentadas. Si tal pavimentación no es posible, se escribirá `Imposible`.

Entrada de ejemplo

```
5
7
1 2 7
1 3 6
3 2 1
1 4 10
3 4 15
3 5 3
4 5 12
4
3
1 3 2
1 4 3
3 4 5
```

Salida de ejemplo

```
20
Imposible
```

Autor: Alberto Verdejo.

17. 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 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 (entre 1 y 10.000) de casas en la comarca y la segunda el número C (entre 1 y 100.000) de conexiones directas entre casas. 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; y 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

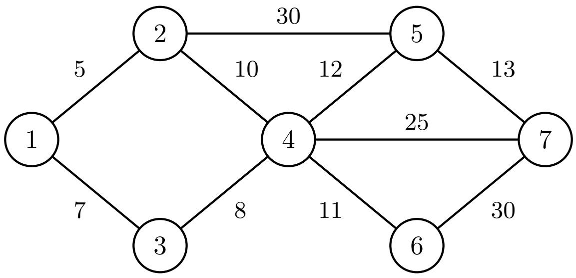
```
35
Imposible
```

Autor: Alberto Verdejo.

18. Camino al cole

Lucas va todos los días en bici al cole. Como le cuesta una barbaridad levantarse por las mañanas, siempre lleva prisa y tiene que ir por el camino más corto. Pero le aburre ir siempre por el mismo camino, por lo que va alternando, eso sí, recorriendo siempre la misma distancia mínima, para no llegar tarde. Como es un poco temerario, no respeta el sentido de circulación de las calles por lo que es capaz de recorrer cualquiera de ellas en ambos sentidos. Ahora se pregunta de cuántas formas distintas puede ir de su casa al colegio sin recorrer ni un centímetro de más.

Por ejemplo, si el siguiente esquema representara el pueblo de Lucas, con 7 intersecciones y 10 calles, donde junto a cada calle aparece su longitud, Lucas viviera en la intersección número 1 y su colegio se encontrara en la intersección 7, entonces Lucas podría ir al colegio de 4 formas distintas recorriendo una distancia de 40 ($1 \to 2 \to 4 \to 5 \to 7$, $1 \to 2 \to 4 \to 7$, $1 \to 3 \to 4 \to 7$ y $1 \to 3 \to 4 \to 5 \to 7$).



Entrada

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

En la primera línea aparece el número N (entre 1 y 10.000) de intersecciones en el pueblo de Lucas, y en la segunda línea aparece el número C (entre 0 y 100.000) de calles (entre intersecciones). A continuación, aparece una línea por cada calle con tres enteros, que indican los números de las intersecciones que une la calle (números entre 1 y N) y su longitud (un valor entre 1 y 100.000). Nunca hay una calle que conecte una intersección consigo misma, ni dos calles que conecten el mismo par de intersecciones.

La casa de Lucas siempre se encuentra en la intersección número 1 y su colegio está situado en la intersección número N.

Salida

Para cada caso de prueba se escribirá una línea con el número de formas distintas de ir desde la casa de Lucas al colegio recorriendo la mínima distancia posible. Este número será 0 si es imposible alcanzar el colegio desde la casa de Lucas utilizando las calles existentes.

Entrada de ejemplo

```
7
10
1 2 5
1 3 7
2 4 10
3 4 8
2 5 30
4 5 12
4 7 25
4 6 11
5 7 13
6 7 30
```

Salida de ejemplo

```
4
```

Autor: Alberto Verdejo.