



Werewolf

En la Prefectura de Ibaraki, Japón hay N ciudades y M carreteras. Las ciudades están numeradas de 0 a $N - 1$ en orden creciente de su población. Cada carretera conecta un par de ciudades diferentes, y puede ser recorrida en ambas direcciones. Puedes viajar desde cualquier ciudad a cualquiera otra ciudad usando una o más de esas carreteras.

Planeas Q viajes, numerados desde 0 hasta $Q - 1$. El viaje i ($0 \leq i \leq Q - 1$) es para ir de la ciudad S_i a la ciudad E_i .

Eres un hombre lobo. Tienes dos formas: **forma humana** y **forma lobo**. Al comienzo de cada viaje te encuentras en forma humana. Al final de cada viaje debes estar en forma lobo. Durante el viaje tienes que **transformarte** (cambiar de forma humana a forma lobo) exactamente una vez. Puedes transformarte únicamente cuando estés en alguna ciudad (posiblemente S_i o E_i).

Vivir como hombre lobo no es fácil. Debes evitar ciudades poco pobladas cuando estés en forma humana y evitar ciudades muy pobladas cuando estés en forma lobo. Para cada viaje i , hay dos límites L_i y R_i ($0 \leq L_i \leq R_i \leq N - 1$) que indican cuales ciudades deben ser evitadas. Más específicamente, debes evitar las ciudades $0, 1, \dots, L_i - 1$ cuando estés en forma humana y evitar las ciudades $R_i + 1, R_i + 2, \dots, N - 1$ cuando estés en forma lobo. Esto quiere decir que en el viaje i puedes transformarte solamente en una de las ciudades $L_i, L_i + 1, \dots, R_i$.

Tu tarea es determinar para cada viaje si es posible ir desde la ciudad S_i a la ciudad E_i , en una manera que satisfaga las restricciones antes mencionadas. El camino que tomes puede tener una longitud arbitraria.

Detalles de la implementación

Debes implementar la siguiente función:

```
int[] check_validity(int N, int[] X, int[] Y, int[] S, int[] E, int[] L, int[] R)
```

- N : el número de ciudades.
- X e Y : arreglos de longitud M . Para cada j ($0 \leq j \leq M - 1$), la ciudad $X[j]$ está directamente conectada a la ciudad $Y[j]$ por una carretera.

- S , E , L , y R : arreglos de longitud Q , representando los viajes.

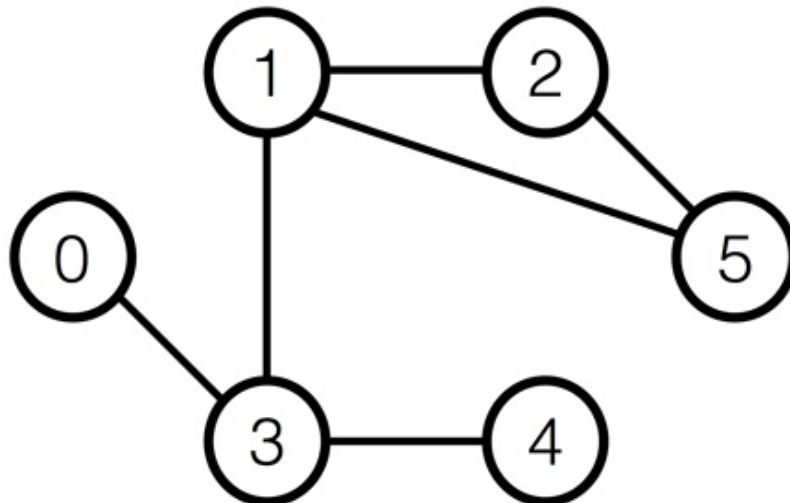
Nota que los valores de M y Q son las longitudes de los arreglos y pueden ser obtenidos como se indica en las notas de implementación.

La función `check_validity` es llamada exactamente una vez por cada caso de prueba. Esta función debe retornar un arreglo de enteros A de longitud Q . El valor de A_i ($0 \leq i \leq Q - 1$) debe ser 1 si es posible ir de la ciudad S_i a la ciudad E_i satisfaciendo las condiciones mencionadas anteriormente, o 0 en otro caso.

Ejemplo

Sean $N = 6$, $M = 6$, $Q = 3$, $X = [5, 1, 1, 3, 3, 5]$, $Y = [1, 2, 3, 4, 0, 2]$, $S = [4, 4, 5]$, $E = [2, 2, 4]$, $L = [1, 2, 3]$, y $R = [2, 2, 4]$.

El evaluador llama `check_validity(6, [5, 1, 1, 3, 3, 5], [1, 2, 3, 4, 0, 2], [4, 4, 5], [2, 2, 4], [1, 2, 3], [2, 2, 4])`.



Para el viaje 0, puedes ir de la ciudad 4 a la ciudad 2 de la siguiente manera:

- Empieza en la ciudad 4 (Estás en forma humana)
- Va a la ciudad 3 (Estás en forma humana)
- Va a la ciudad 1 (Estás en forma humana)
- Se transforma en lobo (Estás en forma lobo)
- Va a la ciudad 2 (Estás en forma lobo)

Para los viajes 1 y 2, no puedes desplazarte entre las ciudades dadas. Así que, tu programa debe retornar `[1, 0, 0]`.

Los archivos `sample-01-in.txt` y `sample-01-out.txt` en el archivo zip corresponden a este ejemplo. También hay otras entradas/salidas ejemplo en el zip.

Restricciones

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- Para cada $0 \leq j \leq M - 1$
 - $0 \leq X_j \leq N - 1$
 - $0 \leq Y_j \leq N - 1$
 - $X_j \neq Y_j$
- Puedes ir de cualquier ciudad a cualquier otra usando las carreteras.
- Cada par de ciudades están conectadas por a lo más una carretera. En otras palabras, para cada $0 \leq j < k \leq M - 1$, $(X_j, Y_j) \neq (X_k, Y_k)$ y $(Y_j, X_j) \neq (X_k, Y_k)$.
- Para cada $0 \leq i \leq Q - 1$
 - $0 \leq L_i \leq S_i \leq N - 1$
 - $0 \leq E_i \leq R_i \leq N - 1$
 - $S_i \neq E_i$
 - $L_i \leq R_i$

Subtareas

1. (7 puntos) $N \leq 100$, $M \leq 200$, $Q \leq 100$
2. (8 puntos) $N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$
3. (34 puntos) $M = N - 1$ y cada ciudad es incidente a máximo dos carreteras (las ciudades están conectadas en una línea)
4. (51 puntos) No hay restricciones adicionales

Grader de ejemplo

El grader de ejemplo lee la entrada en el siguiente formato:

- línea 1: $N M Q$
- línea $2 + j$ ($0 \leq j \leq M - 1$): $X_j Y_j$
- línea $2 + M + i$ ($0 \leq i \leq Q - 1$): $S_i E_i L_i R_i$

El grader de ejemplo imprime el valor de retorno de `check_validity` en el siguiente formato:

- línea $1 + i$ ($0 \leq i \leq Q - 1$): A_i