



# Werewolf

Há  $N$  cidades e  $M$  estradas em Ibaraki Prefecture, Japão. As cidades são numeradas de 0 até  $N - 1$  em ordem crescente da sua população. Cada estrada liga um par de cidades distintas, e pode ser atravessada em ambas as direções. É possível viajar de cada cidade para uma outra qualquer cidade usando uma ou mais destas estradas.

Você planejou  $Q$  viagens, numeradas de 0 até  $Q - 1$ . A viagem  $i$  ( $0 \leq i \leq Q - 1$ ) consiste em viajar da cidade  $S_i$  para a cidade  $E_i$ .

Você é um lobisOMEM. Você tem duas formas: **forma humana** e **forma lobo**. No início de cada viagem, você está em forma humana. No fim de cada viagem, você tem de estar em forma lobo. Durante a viagem você tem de se **transformar** (mudar de forma humana para forma lobo) exatamente uma vez. Pode transformar-se apenas quando está numa cidade (possivelmente  $S_i$  or  $E_i$ ).

Ser um lobisOMEM não é fácil. Você deve evitar cidades com população baixa quando está em forma humana e evitar cidades com população alta quando está em forma lobo. Para cada viagem  $i$  ( $0 \leq i \leq Q - 1$ ), há dois limites  $L_i$  e  $R_i$  ( $0 \leq L_i \leq R_i \leq N - 1$ ) que indicam quais cidades devem ser evitadas. Mais especificamente, deve evitar as cidades  $0, 1, \dots, L_i - 1$  quando estiver em forma humana e evitar as cidades  $R_i + 1, R_i + 2, \dots, N - 1$  quando estiver em forma lobo.

Isto significa que na viagem  $i$ , só pode se transformar numa das cidades  $L_i, L_i + 1, \dots, R_i$ .

A sua tarefa é determinar, para cada viagem, se é possível viajar da cidade  $S_i$  para a cidade  $E_i$  de forma a satisfazer as condições indicadas. O caminho escolhido pode ter comprimento arbitrário.

## Detalhes de implementação

Você deve implementar a seguinte função:

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

- $N$ : o número de cidades.
- $X$  e  $Y$ : arrays de comprimento  $M$ . Para cada  $j$  ( $0 \leq j \leq M - 1$ ), a cidade  $X[j]$  está diretamente ligada à cidade  $Y[j]$  por uma estrada.

- $S$ ,  $E$ ,  $L$ , e  $R$ : arrays de comprimento  $Q$ , representando as viagens.

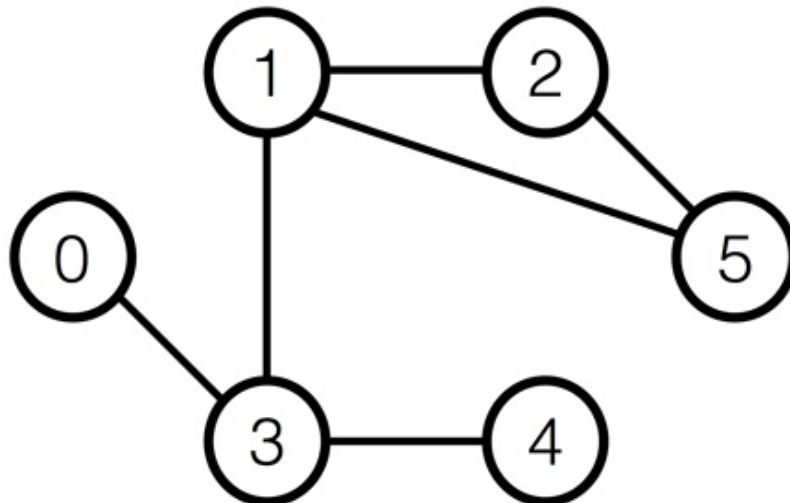
Note que os valores de  $M$  e  $Q$  são os comprimentos dos arrays, e podem ser obtidos como indicado na nota de implementação.

A função `check_validity` é chamada exatamente uma vez por caso de teste. Esta função deve retornar um array  $A$  de inteiros de comprimento  $Q$ . O valor de  $A_i$  ( $0 \leq i \leq Q - 1$ ) deverá ser 1 se a viagem  $i$  é possível satisfazendo as condições mencionadas, ou 0 caso contrário.

## Exemplo

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

O avaliador de exemplo chama `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 a viagem 0, pode viajar da cidade 4 para a cidade 2 da seguinte forma:

- Começa na cidade 4 (você está em forma humana)
- Viaja para a cidade 3 (você está em forma humana)
- Viaja para a cidade 1 (você está em forma humana)
- Transforma-se para a sua forma lobo (você está em forma lobo)
- Viaja para a cidade 2 (você está em forma lobo)

Para as viagens 1 e 2, não é possível viajar entre as cidades dadas.

Assim, o seu programa deve retornar `[1, 0, 0]`.

Os arquivos `sample-01-in.txt` e `sample-01-out.txt` no pacote anexo compactado correspondem a este exemplo. Também estão disponíveis no pacote outros exemplos de input/output.

## Restrições

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- Para cada  $0 \leq j \leq M - 1$ 
  - $0 \leq X_j \leq N - 1$
  - $0 \leq Y_j \leq N - 1$
  - $X_j \neq Y_j$
- É possível viajar de cada cidade para cada outra cidade usando as estradas.
- Cada par de cidades está ligado por no máximo uma estrada. Por outras palavras, para todo  $0 \leq j < k \leq M - 1$ ,  $(X_j, Y_j) \neq (X_k, Y_k)$  e  $(Y_j, X_j) \neq (X_k, Y_k)$ .
- Para cada  $0 \leq i \leq Q - 1$ 
  - $0 \leq L_i \leq S_i \leq N - 1$
  - $0 \leq E_i \leq R_i \leq N - 1$
  - $S_i \neq E_i$
  - $L_i \leq R_i$

## Subtarefas

1. (7 pontos)  $N \leq 100$ ,  $M \leq 200$ ,  $Q \leq 100$
2. (8 pontos)  $N \leq 3\,000$ ,  $M \leq 6\,000$ ,  $Q \leq 3\,000$
3. (34 pontos)  $M = N - 1$  e cada cidade é incidente a no máximo 2 estradas (as cidades estão ligadas em linha)
4. (51 pontos) Nenhuma restrição adicional

## Corretor de exemplo

O corretor de exemplo lê o input no seguinte formato:

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

O corretor de exemplo imprime o valor de retorno de `check_validity` no seguinte formato:

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