



Адам-Бөрү

Жапанда, Ибараки аймагында N шаар жана M жол бар. Шаарлар тургундарынын сандары боюнча өсүүчү тартибинде 0дөн $(N - 1)$ га дейре номурланган. Ар бир жол ар түрдүү эки шаарды байлайт, ар бир жол экөө багытта өткөрүлө алат. Каалаган шаардан башка каалаган шаарга жол(дор) боюнча өтүш мүмкүн. (0дөн $(Q - 1)$ га дейре номурланган) Q саякатты каалайсын. i -инчи саякат ($0 \leq i \leq Q - 1$) S_i -инчи шаардан E_i -инчи шаарга өтүү.

Сен адам-бөрүсүң. Сенин эки түрүң: **адам түрү** жана **бөрү түрү** бар экен. Ар бир саякаттын башында сен адам түрүндө, аягында бөрү түрүндө болосуң. Саякат жүргөндө, сен бир гана жолу **өзгөрүш** (адам түрүнөн бөрү түрүнө өтүш) керек. Бул өзгөрүшүң кандайдыр бир шаарда (алардын ичинде S_i -инчи шаар жана E_i -инчи шаар) болуш керек. Бөрү түрүндө жашоо жеңил эмес. Тажрыйбаңдан, сен адам түрүндө тургундарынын саны аз болгон шаарларга кирүүсүңө болбойт, бөрү түрүндө тургундарынын саны көп болгон шаарларга кирүүсүңө болбойт. Тагыраак, ар бир i -инчи саякат үчүн эки бүтүн L_i жана R_i ($0 \leq L_i \leq R_i \leq N - 1$) саны берилет. i -инчи саякат үчүн, адам түрүндө 0-үнчү, 1-инчи, ..., $(L_i - 1)$ -инчи шаар(лар)га кирүүсүңө болбойт, бөрү түрүндө $(R_i + 1)$ -инчи, $(R_i + 2)$ -инчи, ..., $(N - 1)$ -инчи шаар(лар)га кирүүсүңө болбойт. Ошондуктан, L_i -инчи, $(L_i + 1)$ -инчи, ..., R_i -инчи шаар(лар)дын биринде өзгөрүшүң болуш мүмкүн. Ар бир саякат үчүн, сенин тапшырмаң S_i -инчи шаардан E_i -инчи шаарга айтылган шарттарды канааттандыруучу өтүү болоор-болбосун аныктоо. Саякаттын узундугу каалаган болуш мүмкүн.

Implementation details

You should implement the following function:

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

- N : the number of cities.
- X and Y : arrays of length M . For each j ($0 \leq j \leq M - 1$), the city $X[j]$ is directly connected to the city $Y[j]$ by a road.
- S , E , L , and R : arrays of length Q , representing the trips.

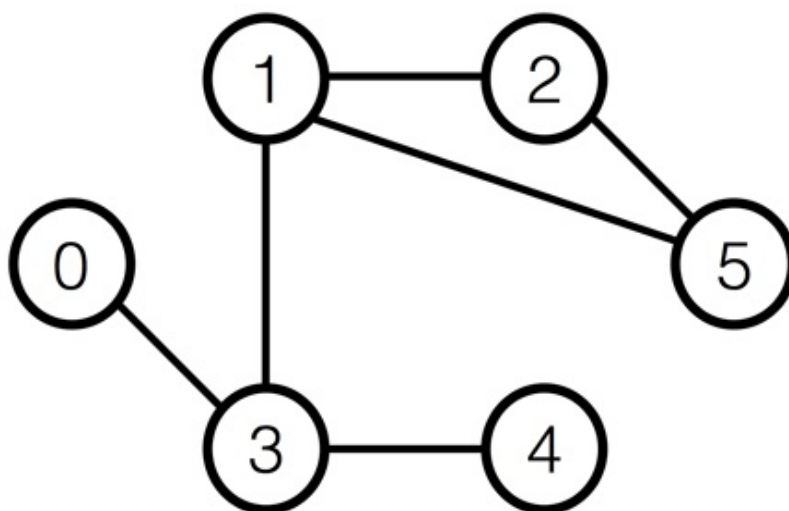
The function `check_validity` is called exactly once for each test case. This function should return an array A of integers of length Q . The value of A_i ($0 \leq i \leq Q - 1$) must

be 1 if it is possible to travel from the city S_i to the city E_i , avoiding the cities $0, 1, \dots, L_i - 1$ when you are in human form, and the cities $R_i + 1, R_i + 2, \dots, N - 1$ when you are in wolf form. Otherwise, it must be 0.

Example

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

The grader calls `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])`.



For the trip 0, you can travel from the city 4 to the city 2 as follows:

- Start at the city 4 (You are in human form)
- Move to the city 3 (You are in human form)
- Move to the city 1 (You are in human form)
- Transform yourself into wolf form (You are in wolf form)
- Move to the city 2 (You are in wolf form)

For the trips 1 and 2, you cannot travel between the given cities.

Hence, your program should return `[1, 0, 0]`.

The files `sample-01-in.txt` and `sample-01-out.txt` in the zipped attachment package correspond to this example. Other sample input/output are also available in the package.

Constraints

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$

- $0 \leq X_j \leq N - 1$ ($0 \leq j \leq M - 1$)
- $0 \leq Y_j \leq N - 1$ ($0 \leq j \leq M - 1$)
- You can travel from any city to any other city by using roads.
- $X_j \neq Y_j$ ($0 \leq j \leq M - 1$)
- $(X_j, Y_j) \neq (X_k, Y_k)$ and $(X_j, Y_j) \neq (Y_k, X_k)$ ($0 \leq j < k \leq M - 1$)
- $0 \leq S_i \leq N - 1$ ($0 \leq i \leq Q - 1$)
- $0 \leq E_i \leq N - 1$ ($0 \leq i \leq Q - 1$)
- $S_i \neq E_i$ ($0 \leq i \leq Q - 1$)
- $0 \leq L_i \leq R_i \leq N - 1$ ($1 \leq i \leq Q - 1$)
- $L_i \leq S_i$ ($0 \leq i \leq Q - 1$)
- $E_i \leq R_i$ ($0 \leq i \leq Q - 1$)

Subtasks

1. (7 points) $N \leq 100$, $M \leq 200$, $Q \leq 100$
2. (8 points) $N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$
3. (34 points) $M = N - 1$ and no city is directly connected to more than 2 cities (the cities are connected in a line)
4. (51 points) No additional constraints

Sample grader

The sample grader reads the input in the following format:

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

The sample grader prints the return value of `check_validity` in the following format:

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