# Werewolf

There are $N$ cities and $M$ roads in Ibaraki Prefecture, Japan. Cities are numbered from 0 through $N-1$ in the increasing order of their population. Each road connects a pair of distinct cities, and can be traveled in both directions. You can travel from any city to any other city by using one or more of these roads.

You planned $Q$ trips, numbered from 0 through $Q-1$. The trip $i$ ($0 \leq i \leq Q-1$) is to travel from the city $S_i$ to the city $E_i$.

You are a werewolf. You have two forms: **human form** and **wolf form**. At the beginning of each trip you are in human form. At the end of each trip, you must be in wolf form. During the trip you have to **transform** (change from human form to wolf form) exactly once. You can transform only when you are in some city (possibly $S_i$ or $E_i$).

Living as a werewolf is not easy. You must avoid low-populated cities when you are in human form, and avoid highly-populated cities when you are in wolf form. For each trip $i$ ($0 \leq i \leq Q-1$), there are two thresholds $L_i$ and $R_i$ ($0 \leq L_i \leq R_i \leq N-1$) that indicate which cities must be avoided. More specifically, you must avoid the cities $0, 1, \ldots, L_i-1$ when you are in human form, and must avoid the cities $R_i+1, R_i+2, \ldots, N-1$ when you are in wolf form. This means in the trip $i$, you can only transform in one of the cities $L_i, L_i+1, \ldots, R_i$.

Your task is to determine, for each trip, whether it is possible to travel from the city $S_i$ to the city $E_i$ in a way that satisfies the aforementioned constraints. The route you take can have an arbitrary length.

## 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.

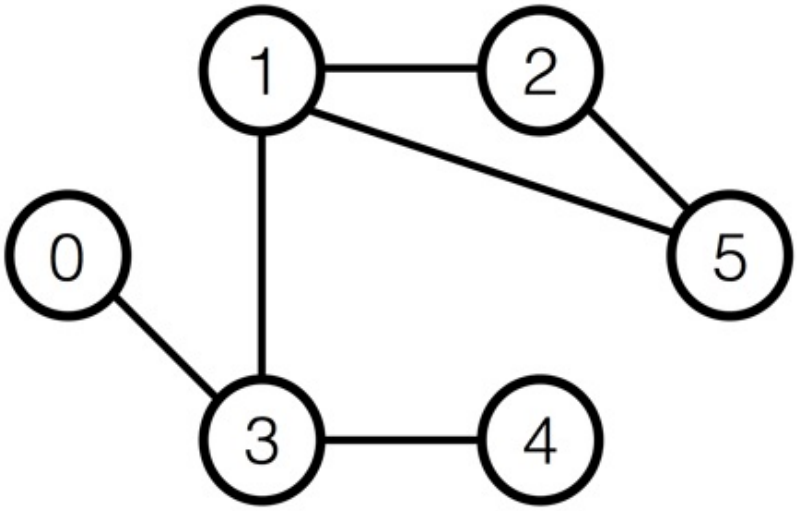
Note that the values of $M$ and $Q$ are the lengths of the arrays, and can be obtained as indicated in the implementation notice.

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 the trip $i$ is possible while satisfying the aforementioned conditions, or `0` otherwise.

## 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. This package also contains another pair of sample input/output files.

## Constraints

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- For each $0 \leq j \leq M - 1$
  - $0 \leq X_j \leq N - 1$
  - $0 \leq Y_j \leq N - 1$
  - $X_j \neq Y_j$
- You can travel from any city to any other city by using roads.
- Each pair of cities are directly connected by at most one road. In other words, for all $0 \leq j < k \leq M - 1$, $(X_j, Y_j) \neq (X_k, Y_k)$ and $(Y_j, X_j) \neq (X_k, Y_k)$.
- For each $0 \leq i \leq Q - 1$
  - $0 \leq L_i \leq S_i \leq N - 1$
  - $0 \leq E_i \leq R_i \leq N - 1$
  - $S_i \neq E_i$
  - $L_i \leq R_i$

# 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 each city is incident to at most $2$ roads (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 $1 + i$ ($0 \leq i \leq Q - 1$): $A_i$