



Werewolf

Quận Ibaraki ở Nhật Bản có N thành phố và M con đường. Các thành phố được đánh số từ 0 đến $N - 1$ theo thứ tự tăng dần về số lượng dân trong thành phố. Mỗi con đường nối một cặp hai thành phố khác nhau, và có thể di chuyển theo cả hai hướng. Bạn có thể sử dụng một hoặc nhiều con đường này để di chuyển từ một thành phố bất kì đến một thành phố bất kì khác.

Bạn lên kế hoạch gồm Q chuyến đi, được đánh số từ 0 đến $Q - 1$. Chuyến đi thứ i ($0 \leq i \leq Q - 1$) sẽ di chuyển từ thành phố S_i đến thành phố E_i .

Bạn là người sói. Bạn có hai hình dáng: **Hình dáng người** và **hình dáng sói**. Tại thời điểm bắt đầu của mỗi chuyến đi, bạn ở hình dáng người. Kết thúc chuyến đi, bạn bắt buộc phải ở hình dáng sói. Trong chuyến đi, bạn phải **chuyển đổi** (chuyển từ hình dáng người sang hình dáng sói) đúng một lần. Bạn chỉ có thể chuyển đổi khi bạn đang ở một thành phố nào đó (kể cả S_i hoặc E_i).

Cuộc sống của người sói không dễ dàng. Bạn phải tránh ở hình dáng người khi ở các thành phố ít dân và tránh ở hình dáng sói khi ở các thành phố đông dân. Với mỗi chuyến đi i ($0 \leq i \leq Q - 1$), có hai ngưỡng L_i và R_i ($0 \leq L_i \leq R_i \leq N - 1$) chỉ ra các thành phố nào bạn phải tránh. Cụ thể là, bạn không được ở hình dáng người khi ở các thành phố $0, 1, \dots, L_i - 1$, và không được ở hình dáng sói khi ở các thành phố $R_i + 1, R_i + 2, \dots, N - 1$. Điều đó có nghĩa là trong chuyến đi thứ i , bạn chỉ được chuyển đổi hình dáng ở một trong các thành phố $L_i, L_i + 1, \dots, R_i$.

Nhiệm vụ của bạn là xác định xem đối với mỗi chuyến đi có tồn tại hay không cách di chuyển từ thành phố S_i đến E_i thỏa mãn tất cả các điều kiện ở trên. Chuyến đi của bạn có thể có độ dài bất kì.

Chi tiết cài đặt

Bạn cần cài đặt hàm sau đây:

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

- N : Số lượng thành phố.
- X và Y : Các mảng với độ dài M . Với mỗi j ($0 \leq j \leq M - 1$), thành phố $X[j]$ được nối trực tiếp với thành phố $Y[j]$ bởi một con đường.

- S, E, L, và R: Các mảng với độ dài Q , biểu diễn các chuyển đi.

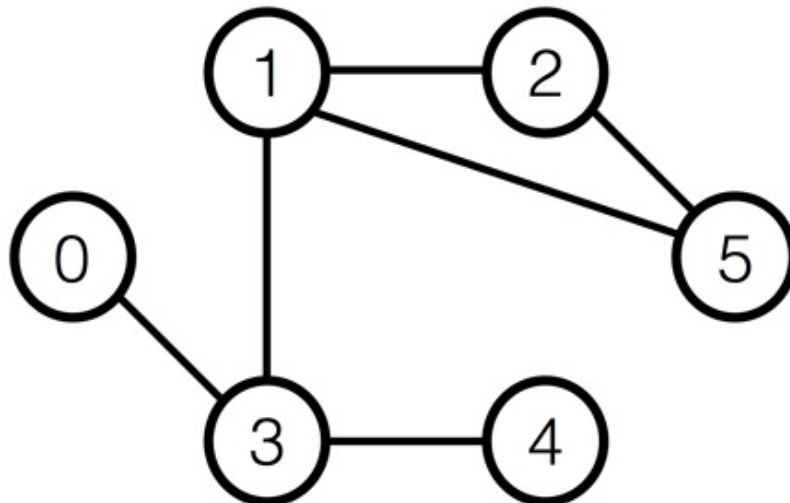
Chú ý, các giá trị của M và Q là độ dài của các mảng, và cách nhận độ dài của mảng được mô tả trong phần Lưu ý cài đặt.

Hàm `check_validity` được gọi đúng một lần cho mỗi test case. Hàm này cần trả về một mảng A các số nguyên với độ dài Q . Giá trị của A_i ($0 \leq i \leq Q - 1$) phải là 1 nếu chuyển đi thứ i có thể thực hiện được thỏa mãn các điều kiện đã nêu, hoặc 0 nếu trái lại.

Ví dụ

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

Chương trình chấm gọi hàm `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])`.



Với chuyển đi 0, bạn có thể di chuyển từ thành phố 4 đến thành phố 2 như sau:

- Bắt đầu tại thành phố 4 (Bạn ở hình dáng người)
- Di chuyển đến thành phố 3 (Bạn ở hình dáng người)
- Di chuyển đến thành phố 1 (Bạn ở hình dáng người)
- Chuyển đổi từ hình dáng người sang hình dáng sói (Bạn ở hình dáng sói)
- Di chuyển đến thành phố 2 (Bạn ở hình dáng sói)

Với chuyển đi 1 và 2, bạn không thể di chuyển giữa các thành phố đã cho.

Do đó, chương trình của bạn sẽ trả lại $[1, 0, 0]$.

Các file `sample-01-in.txt` và `sample-01-out.txt` trong gói file nén zip đính kèm tương ứng với ví dụ này. Gói đính kèm này còn chứa một cặp file ví dụ input/output khác.

Hạn chế

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- Với mỗi $0 \leq j \leq M - 1$
 - $0 \leq X_j \leq N - 1$
 - $0 \leq Y_j \leq N - 1$
 - $X_j \neq Y_j$
- Bạn có thể di chuyển từ một thành phố bất kì đến một thành phố bất kì khác bằng cách sử dụng các con đường.
- Mỗi cặp thành phố được nối trực tiếp với nhau bởi nhiều nhất một con đường. Nói cách khác, với mọi $0 \leq j < k \leq M - 1$, $(X_j, Y_j) \neq (X_k, Y_k)$ và $(Y_j, X_j) \neq (X_k, Y_k)$.
- Với mỗi $0 \leq i \leq Q - 1$
 - $0 \leq L_i \leq S_i \leq N - 1$
 - $0 \leq E_i \leq R_i \leq N - 1$
 - $S_i \neq E_i$
 - $L_i \leq R_i$

Subtasks

1. (7 điểm) $N \leq 100$, $M \leq 200$, $Q \leq 100$
2. (8 điểm) $N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$
3. (34 điểm) $M = N - 1$ và mỗi thành phố là đầu mút của nhiều nhất là 2 con đường (các thành phố được nối thành một đường).
4. (51 điểm) Không có hạn chế bổ sung

Trình chấm mẫu

Trình chấm mẫu đọc dữ liệu đầu vào theo khuôn dạng sau:

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

Trình chấm mẫu in ra và trả lại giá trị của hàm `check_validity` theo khuôn dạng sau:

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