



Farkasember

Ibarakiban N város és M közvetlen út található. A városokat a népességük szerint növekvő sorrendben 0 -tól $N - 1$ -ig azonosítjuk. Minden közvetlen út kétirányú, két különböző várost köt össze. Bármely két város között létezik útvonal.

Q útvonalat terveztél, az i . útvonal az S_i városból az E_i városba vezet.

Farkasember vagy, ezért **emberi** és **farkas** formában is lehetsz az út során. Kezdetben ember vagy, a végén farkasnak kell lenned! Az út során pontosan egyszer kell átváltoznod farkassá, valamelyik városban!

Az i . utazásnál választottál L_i és R_i egész számot, amelyekre teljesül, hogy $0 \leq L_i \leq R_i \leq N - 1$. Az i . utazás során el akarsz kerülni a $0, 1, \dots, L_i - 1$ városokat, amikor ember vagy, illetve a $R_i + 1, R_i + 2, \dots, N - 1$ városokat, ha farkas vagy. Ez azt jelenti, hogy az $L_i, L_i + 1, \dots, R_i$ városok valamelyikében kell átváltoznod farkassá.

Határozd meg minden utazásra, hogy lehetséges-e eljutni S_i városból E_i városba, a feltételeknek megfelelő, tetszőleges hosszú útvonalon!

Megvalósítás

A következő függvényt kell megvalósítanod.

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

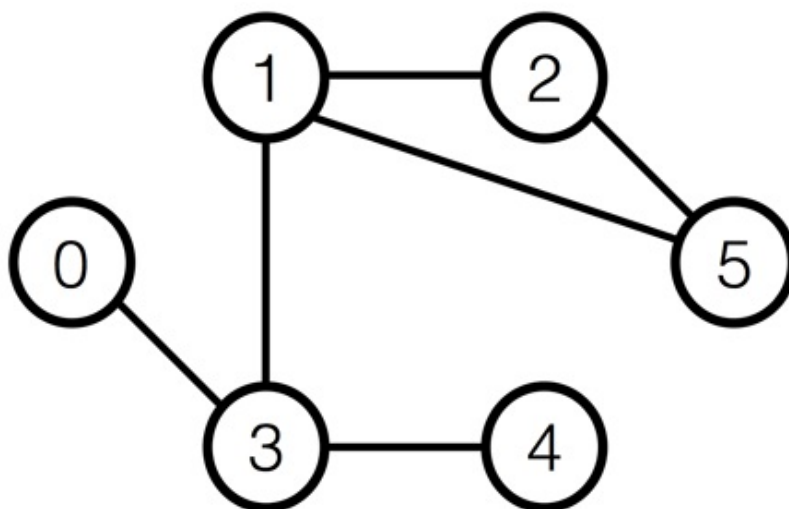
- N : a városok száma.
- X és Y : M elemű vektor (M lekérdezhető a vektor hosszaként).
- Minden j -re ($0 \leq j \leq M - 1$), $X[j]$ -t közvetlen út köti össze $Y[j]$ -vel.
- S , E , L , és R : Q elemű vektorok, az utazásokat írják le.

A `check_validity` függvényt tesztelésenként egyszer hívják meg. A függvény az A Q elemű vektort adja eredményül! Az A_i értéke ($0 \leq i \leq Q - 1$) legyen 1 , ha az i . útvonal olyan, hogy az S_i . városból az E_i . városba vezet, és elkerülheti a $0, 1, \dots, L_i - 1$ városokat emberként, az $R_i + 1, R_i + 2, \dots, N - 1$ városokat pedig farkasként. Egyébként legyen 0 értékű!

Példa

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

Az értékelő hívása: `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])`.



A 0. utazás során a 4. városból a 2. városba kell eljutni, ami így lehetséges:

- Indulás 4-ből (emberként)
- Átlépés 3-ba (emberként)
- Átlépés 1-be (emberként)
- Átváltozol farkassá
- Átlépés 2-be (farkasként)

Az 1. és a 2. utazás nem hajtható végre a feltételeknek megfelelően.

Így a függvényed eredménye `[1, 0, 0]` legyen!

A tömörített állományban a `sample-01-in.txt` és a `sample-01-out.txt` tartalmazza a fenti példát. Más példák is vannak benne.

Korlátok

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- $0 \leq X_j \leq N - 1$ ($0 \leq j \leq M - 1$)
- $0 \leq Y_j \leq N - 1$ ($0 \leq j \leq M - 1$)
- Bármely városból bármely másikba el lehet jutni.
- $X_j \neq Y_j$ ($0 \leq j \leq M - 1$)
- $(X_j, Y_j) \neq (X_k, Y_k)$ és $(X_j, Y_j) \neq (Y_k, X_k)$ ($0 \leq j < k \leq M - 1$)
- $0 \leq S_i \leq N - 1$ ($0 \leq i \leq Q - 1$)
- $0 \leq E_i \leq N - 1$ ($0 \leq i \leq Q - 1$)

- $S_i \neq E_i$ ($0 \leq i \leq Q - 1$)
- $0 \leq L_i \leq R_i \leq N - 1$ ($1 \leq i \leq Q - 1$)
- $L_i \leq S_i$ ($0 \leq i \leq Q - 1$)
- $E_i \leq R_i$ ($0 \leq i \leq Q - 1$)

Részfeladatok

1. (7 pont) $N \leq 100$, $M \leq 200$, $Q \leq 100$
2. (8 pont) $N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$
3. (34 pont) $M = N - 1$ és minden városnak legfeljebb 2 szomszédja van (tehát egy láncot alkotnak)
4. (51 pont) nincs további feltétel

Minta értékelő

A bemenetet a következő formában olvassa:

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

A függvényed eredményét a következőképpen írja ki:

- Az $1 + i$. sor ($0 \leq i \leq Q - 1$): A_i