



Werewolf

U Ličko-senjskoj županiji nalazi se N gradova i M cesta. Gradovi su numerirani od 0 do $N - 1$ prema broju stanovnika (od najmanjeg do najvećeg). Svaka je cesta dvosmjerna te povezuje dva različita grada. Također, cestovna mreža je takva da povezuje svaki par gradova.

Mirko, inače lokalni vukodlak, isplanirao je Q putovanja koja je numerirao brojevima od 0 do $Q - 1$. Na i -tom ($0 \leq i \leq Q - 1$) putovanju, Mirko će putovati od grada S_i do grada E_i .

Opće je poznato da vukodlaci mogu poprimiti dva oblika, **ljudski oblik** i **vučji oblik**. Mirko se pobrinuo da se na početku putovanja nalazi u ljudskom obliku, a na kraju putovanja mora se nalaziti u vučjem obliku. Mirko se tijekom putovanja mora **transformirati** (promijeniti oblik iz čovjeka u vuka) točno jednom i tada se mora nalaziti u nekom gradu (uključivo S_i i E_i).

Biti vukodlak nije šala. Mirkovo iskustvo nalaže da je bolje izbjegavati slabo naseljene gradove kada se nalazi u ljudskom obliku, dok će u vučjem obliku izbjegavati gradove s velikim brojem stanovnika. Preciznije, za i -to putovanje Mirko je odlučio izbjegavati gradove $0, 1, \dots, L_i - 1$ kada se nalazi u ljudskom obliku, odnosno odlučio je izbjegavati gradove $R_i + 1, R_i + 2, \dots, N - 1$ kada se nalazi u vučjem obliku što implicira da transformaciju mora napraviti u jednom od gradova $L_i, L_i + 1, \dots, R_i$. Za gradove L_i i R_i vrijedi ($0 \leq L_i \leq R_i \leq N - 1$).

Vaš je zadatak pomoći Mirku te za svako putovanje odrediti je li ono izvedivo, odnosno, je li moguće putovati iz grada S_i do grada E_i tako da su pritom ispoštovana prethodno spomenuta ograničenja. Primijetite da Mirkovo putovanje može biti proizvoljne duljine.

Implementacijski detalji

Implementirajte sljedeću funkciju:

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

- N : broj gradova Ličko-senjske županije.
- X and Y : polja duljine M . Za svaki j ($0 \leq j \leq M - 1$), grad $X[j]$ je povezan cestom s gradom $Y[j]$.

- S , E , L i R : polja duljine Q koja predstavljaju Mirkova putovanja.

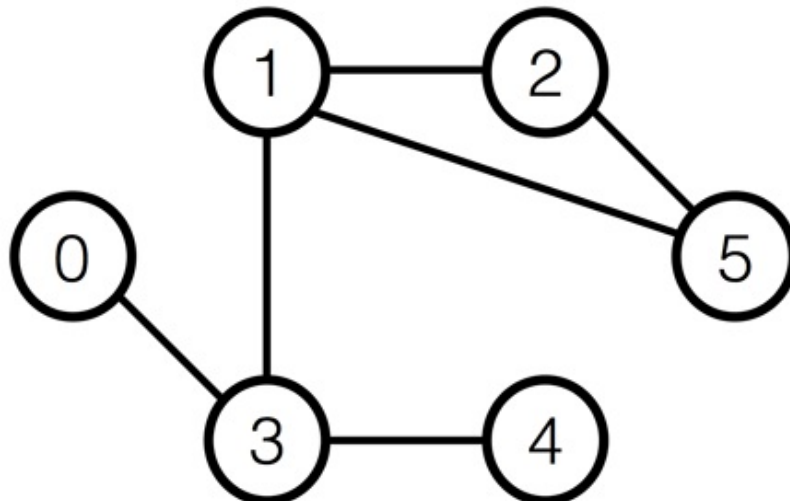
Primijetite da vrijednosti M i Q odgovaraju duljinama polja i mogu se odrediti shodno dokumentu o implementacijskim uputama.

Funkcija `check_validity` poziva se točno jednom za svaki test podatak. Ova funkcija vraća polje cijelih brojeva A duljine Q . Vrijednost elementa A_i ($0 \leq i \leq Q - 1$) mora biti 1 ako je moguće putovati iz grada S_i do grada E_i poštujući ograničenja navedena u tekstu zadatka. U protivnom, vrijednost elementa A_i treba biti 0.

Primjer

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

Grader poziva `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])`.



Za putovanje 0, Mirko može putovati od grada 4 do grada 2 na sljedeći način:

- Započinje u gradu 4 (u ljudskom obliku)
- Putuje u grad 3 (u ljudskom obliku)
- Putuje u grad 1 (u ljudskom obliku)
- Transformira se u vučji oblik.
- Putuje u grad 2 (u vučjem obliku)

Putovanja 1 i 2 nije moguće ostvariti.

Shodno tome, vaš program treba vratiti `[1, 0, 0]`.

Datoteke `sample-01-in.txt` i `sample-01-out.txt` u zapakiranom privitku odgovaraju ovom primjeru. U privitku se također nalaze i neki drugi ogledni primjeri.

Ograničenja

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- Za svaki $0 \leq j \leq M - 1$
 - $0 \leq X_j \leq N - 1$
 - $0 \leq Y_j \leq N - 1$
 - $X_j \neq Y_j$
- Mirko može putovati između bilo koja dva grada koristeći ceste.
- Svaki par gradova direktno je povezan s najviše jednom cestom, odnosno, za sve $0 \leq j < k \leq M - 1$ vrijedi $(X_j, Y_j) \neq (X_k, Y_k)$ i $(Y_j, X_j) \neq (X_k, Y_k)$.
- Za svaki $0 \leq i \leq Q - 1$
 - $0 \leq L_i \leq S_i \leq N - 1$
 - $0 \leq E_i \leq R_i \leq N - 1$
 - $S_i \neq E_i$
 - $L_i \leq R_i$

Podzadaci

1. (7 bodova) $N \leq 100$, $M \leq 200$, $Q \leq 100$
2. (8 bodova) $N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$
3. (34 boda) $M = N - 1$ te niti jedan grad nije povezan s više od 2 grada (gradovi čine lanac).
4. (51 bod) Nema dodatnih ograničenja

Ogledni *grader*

Ogledni *grader* prihvaća ulaz u sljedećem obliku:

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

Ogledni *grader* ispisuje izlaznu vrijednost funkcije `check_validity` u sljedećem obliku:

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