Werewolf

U prefekturi Ibaraki postoji N gradova i M puteva. Gradovi su označeni brojevima od 0 do $N-1$ u rastućem redosledu po broju stanovnika. Svaki put spaja dva različita grada i dvosmeran je. Moguće je stići od bilo kog do bilo kog drugog grada pomoću ovih puteva.

Pavle je isplanirao Q putovanja, koja su označena brojevima od 0 do $Q-1$. Putovanje sa rednim brojem i ($0 \leq i \leq Q-1$) se sastoji od toga da se krene od grada S_i i dođe do grada E_i.

Pavle je vukodlak. Ovo u prevodu znači da on ima dve forme: **Pavle formu** i **Šebez formu**. Na početku svakog putovanja Pavle je u svojoj ljudskoj (Pavle) formi. Na kraju svakog putovanja, Pavle mora biti u Šebez formi. Ovo znači da će Pavle nekad u toku putovanja da se **transformiše** (da se pretvori iz Pavla u Šebeza) tačno jednom i to mora da se desi u nekom gradu (možda S_i ili E_i).

Živeti kao vukodlak nija lako. Neophodno je da Pavle zaobilazi gradove sa malom populacijom kada je u Pavle formi i da zaobilazi gradove sa velikom populacijom kada je u Šebez formi. Tačnije, za svako putovanje i, postoje brojevi (ograničenja) L_i i R_i koji zadovoljavaju $0 \leq L_i \leq R_i \leq N-1$. Ovo znači da kod putovanja sa rednim brojem i Pavle mora zaobići gradove $0, 1, \ldots, L_i - 1$ kada je u Pavle formi i mora zaobići gradove $R_i + 1, R_i + 2, \ldots, N-1$ kada je u Šebez formi. Dakle, u putovanju i Pavle može da se transformiše jedino u jednom od gradova $L_i, L_i + 1, \ldots, R_i$.

Za svako putovanje, vaš zadatak je da odredite da li je moguće da Pavle otputuje od grada S_i do grada E_i tako da budu ispoštovana gore pomenuta ograničenja. Dužine Pavlovih putovanja (broj puteva na njima) mogu biti proizvoljno velike.

Detalji implementacije

Potrebno je da implementirate sledeću funkciju:

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

- `N`: broj gradova.
- `X` i `Y`: nizovi dužine M. Za svako j ($0 \leq j \leq M-1$), grad `X[j]` je direktno povezan sa gradom `Y[j]` jednim putem.

- `S`, `E`, `L` i `R`: nizovi dužine Q koji predstavljaju putovanja.

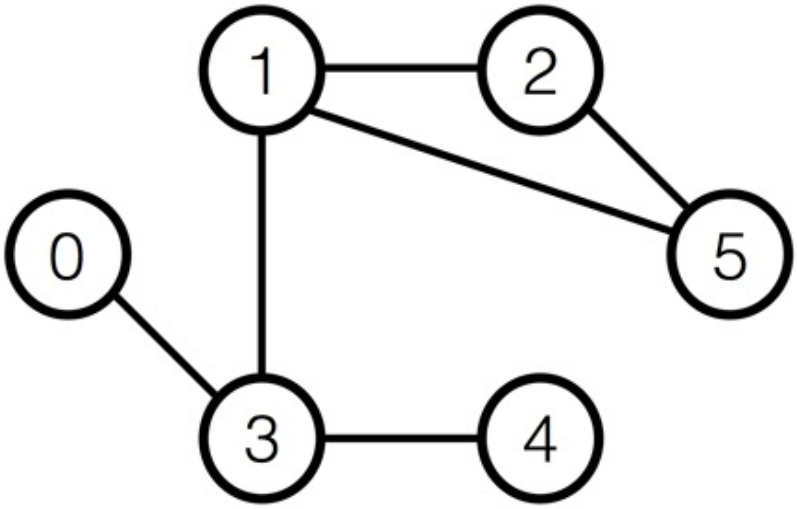
Obratite pažnju, vrednosti M i Q su dužine nizova i mogu se dobiti onako kako je navedeno u napomeni o implementaciji.

Funkcija `check_validity` se zove tačno jednom za svaki test primer. Ova funkcija treba da vrati niz A celih brojeva dužine Q. Vrednost A_i $(0 \leq i \leq Q-1)$ mora biti jednaka 1 ako je moguće da Pavle otputuje od grada S_i do grada E_i zaobilazeći gradove $0, 1, \ldots, L_i - 1$ dok je u Pavle formi, i gradove $R_i + 1, R_i + 2, \ldots, N - 1$ dok je u Šebez formi. U suprotnom, ova vrednost mora biti jednaka 0.

Primer

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 $R = [2, 2, 4]$.

Grejder zove funkciju `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 pod rednim brojem 0, Pavle može da otputuje od grada 4 do grada 2 na sledeći način:

- Pavle započinje u gradu 4 (Trenutno je u Pavle formi)
- Pomeri se u grad 3 (I dalje je u Pavle formi)
- Pomeri se u grad 1 (Još uvek je u Pavle formi)
- Transformiše se u Šebeza (Sad je u Šebez formi)
- Pomeri se u grad 2 (I sada je u Šebez formi)

Za putovanja pod rednim brojem 1 i 2 nije moguće doći od početnog do krajnjeg grada.

Dakle, Vaša funkcija treba da vrati niz $[1, 0, 0]$.

Fajlovi `sample-01-in.txt` i `sample-01-out.txt` u zipovanom dodatku odgovaraju ovom primeru. Drugi primeri ulaza/izlaza su takođe dostupni u ovom dodatku.

Ograničenja

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- Za svako $0 \leq j \leq M - 1$
  - $0 \leq X_j \leq N - 1$
  - $0 \leq Y_j \leq N - 1$
  - $X_j \neq Y_j$
- Moguće je putovati od bilo kog do bilo kog drugog grada koristeći date puteve.
- Svaki par gradova je direktno povezan najviše jednim putem. Drugim rečima, za svako $0 \leq j < k \leq M - 1$, važi $(X_j, Y_j) \neq (X_k, Y_k)$ i $(Y_j, X_j) \neq (X_k, Y_k)$.
- Za svako $0 \leq i \leq Q - 1$
  - $0 \leq L_i \leq S_i \leq N - 1$
  - $0 \leq E_i \leq R_i \leq N - 1$
  - $S_i \neq E_i$
  - $L_i \leq R_i$

Podzadaci

1. (7 poena) $N \leq 100$, $M \leq 200$, $Q \leq 100$
2. (8 poena) $N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$
3. (34 poena) $M = N - 1$ i nijedan grad nije direktno povezan sa više od 2 grada (gradovi su povezani u liniju)
4. (51 poen) Bez dodatnih ograničenja

Priloženi grejder

Priloženi grejder učitava ulaz u sledećem formatu:

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

Priloženi grejder štampa povratnu vrednost funkcije `check_validity` u sledećem formatu:

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