



# Volkodlak

V prefekturi Ibaraka se nahaja  $N$  mest in  $M$  cest. Mesta so oštevilčena od 0 do  $N - 1$  v naraščajočem vrstnem redu glede na število prebivalstva. Vsaka cesta povezuje par različnih mest in po njej se lahko potuje v obe smeri. Iz poljubnega mesta je možno preko ene ali več cest potovati v poljubno drugo mesto.

Načrtuješ  $Q$  potovanj, oštevilčenih od 0 do  $Q - 1$ . Pri  $i$ -tem potovanju ( $0 \leq i \leq Q - 1$ ) potuješ od mesta  $S_i$  do mesta  $E_i$ .

Si volkodlak in zavzemaš dve obliki: **človeško obliko** in **volčjo obliko**. Vsako potovanje pričneš v človeški obliki, zaključiš pa v volčji. Med potovanjem se moraš **preoblikovati** (spremeniti se iz človeške oblike v volčjo) natanko enkrat. Preoblikuješ se lahko le takrat, ko si v nekem mestu (morda  $S_i$  ali  $E_i$ ).

Živeti kot volkodlak ni enostavno. V človeški obliki se moraš izogibati redko poseljenih mest, v volčji pa gosto poseljenih. Za vsako potovanje  $i$  ( $0 \leq i \leq Q - 1$ ) sta podana pragova  $L_i$  in  $R_i$  ( $0 \leq L_i \leq R_i \leq N - 1$ ), ki označujeta, katerih mest se moraš izogniti. Natančneje, v človeški obliki se moraš izogniti mest  $0, 1, \dots, L_i - 1$ , v volčji pa mest  $R_i + 1, R_i + 2, \dots, N - 1$ . To pomeni, da se pri  $i$ -tem potovanju lahko preoblikuješ v enem izmed mest  $L_i, L_i + 1, \dots, R_i$ .

Tvoja naloga je, da za vsako potovanje ugotoviš, ali je možno potovati iz mesta  $S_i$  do mesta  $E_i$  na način, ki upošteva predstavljene omejitve. Pot, ki jo ubereš, je lahko poljubne dolžine.

## Podrobnosti implementacije

Implementiraj naslednjo funkcijo:

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

- $N$ : število mest.
- $X$  and  $Y$ : polji dolžine  $M$ . Za vsak  $j$  ( $0 \leq j \leq M - 1$ ) je mesto  $X[j]$  neposredno povezano z mestom  $Y[j]$  s cesto.
- $S$ ,  $E$ ,  $L$ , and  $R$ : polja dolžine  $Q$ , ki predstavljajo potovanja.

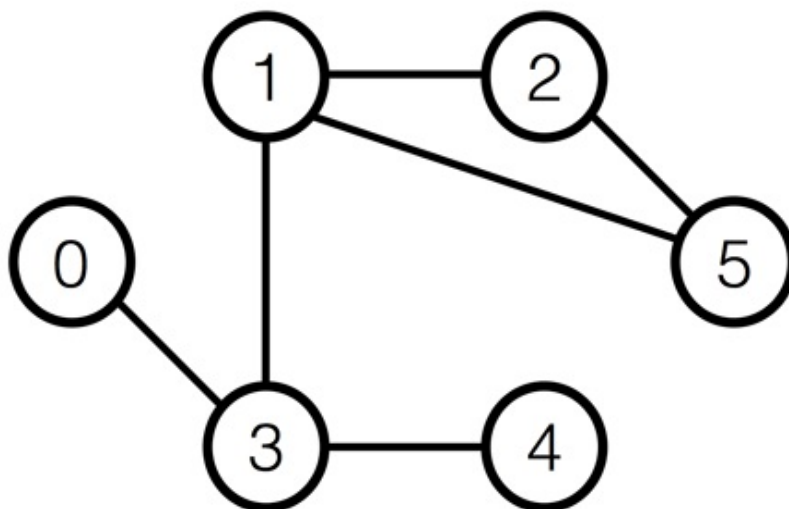
Upoštevaj, da sta vrednosti  $M$  in  $Q$  dolžini polj in da ju je moč pridobiti, kot je navedeno v implementacijskih opombah.

Funkcija `check_validity` se pokliče natanko enkrat za vsak testni primer. Funkcija naj vrne polje celih števil  $A$  z dolžino  $Q$ . Vrednost  $A_i$  ( $0 \leq i \leq Q - 1$ ) mora biti 1, če je pri potovanju  $i$  možno potovati iz mesta  $S_i$  v mesto  $E_i$  tako, da zadostimo zgornjim pogojem, in 0 sicer.

## Primer

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

Ocenjevalnik pokliče `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])`.



Pri potovanju 0 lahko potuješ iz mesta 4 v mesto 2 na naslednji način:

- Začneš v mestu 4 (si v človeški obliki),
- prestaviš se v mesto 3 (si v človeški obliki),
- prestaviš se v mesto 1 (si v človeški obliki),
- preoblikuješ se v volčjo obliko (si v volčji obliki),
- prestaviš se v mesto 2 (si v volčji obliki).

Pri potovanjih 1 in 2 ne moreš potovati med podanimi mesti.

Zatorej tvoj program vrne `[1, 0, 0]`.

Datoteki `sample-01-in.txt` in `sample-01-out.txt` v pripetem paketu zip predstavljata opisani primer. V paketu je na voljo še en primer vhoda/izhoda.

## Omejitve

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- Za vsak  $0 \leq j \leq M - 1$ 
  - $0 \leq X_j \leq N - 1$
  - $0 \leq Y_j \leq N - 1$
  - $X_j \neq Y_j$
- Potuješ lahko iz poljubnega mesta v poljubno drugo mesto preko cest.
- Vsak par mest je neposredno povezan z največ eno cesto. Torej za vse  $0 \leq j < k \leq M - 1$  velja  $(X_j, Y_j) \neq (X_k, Y_k)$  in  $(Y_j, X_j) \neq (X_k, Y_k)$ .
- Za vsak  $0 \leq i \leq Q - 1$ 
  - $0 \leq L_i \leq S_i \leq N - 1$
  - $0 \leq E_i \leq R_i \leq N - 1$
  - $S_i \neq E_i$
  - $L_i \leq R_i$

## Podnaloge

1. (7 točk)  $N \leq 100$ ,  $M \leq 200$ ,  $Q \leq 100$
2. (8 točk)  $N \leq 3\,000$ ,  $M \leq 6\,000$ ,  $Q \leq 3\,000$
3. (34 točk)  $M = N - 1$  in vsako mesto se stika z največ dvema cestama (mesta so povezana v liniji).
4. (51 točk) Ni dodatnih omejitev.

## Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod naslednje oblike:

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

Vzorčni ocenjevalnik izpiše povratno vrednost funkcije `check_validity` v naslednji obliki:

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