



Vlkolak

Krásna Slovenská Príroda sa skladá z N miest, ktoré si pre jednoduchosť označíme číslami od 0 po $N - 1$ podľa počtu obyvateľov – mesto číslo 0 má obyvateľov najmenej, mesto číslo $N - 1$ najviac. Medzi týmito mestami vedie M obojsmerných ciest, každá spája práve dve mestá. Pre Slovensko navyše platí, že každú dvojicu miest spája nejaká postupnosť ciest.

Na prázdniny máš naplánovaných Q výletov, ktoré máš očíslované od 0 po $Q - 1$. Na výlete číslo i sa chceš presunúť z mesta S_i do mesta E_i . Posledný školský deň ťa však v lese pohryzol vlk a z teba sa stal obávaný vlkolak. A plány na leto boli v keli. Iba že by nie...

Ako vlkolak máš dve formy – buď si človek alebo si vlk. Každý výlet začínaš v ľudskej forme. Počas každého výletu musíš svoj výzor zmeniť a výlet ukončiť ako majestátne chlpaté monštrum. Bohužiaľ, zmeniť podobu vieš len raz za celý výlet a túto zmenu musíš spraviť počas pobytu v niektorom meste. (Spln mesiaca na teba nefunguje, ale okružle pouličné lampy áno.) Zmenu podoby môžeš spraviť aj na začiatku alebo konci celého výletu – t.j. aj skôr než opustíš mesto S_i , aj po tom ako pricestuješ do mesta E_i .

To ale stále nie je všetko. Ak sa ako vlk objavíš v nejakom veľmi zaľudnenom mieste, ľudia začnú chytať vidly a zapalovať fakle. A naopak, v totálne vyludnených mestách sa človek nudí, vtedy je väčšia sranda byť vlkom a ponaháňať nejakého úbohého sedliaka. Pre každý výlet si si preto zvolil(a) dve hodnoty L_i a R_i , ktoré spĺňajú nerovnosť $0 \leq L_i \leq R_i \leq N - 1$. Tieto hodnoty znamenajú, že:

- počas i -teho výletu nesmieš byť v mestách $0, 1 \dots L_i - 1$ ako človek
- počas i -teho výletu nesmieš byť v mestách $R_i + 1, R_i + 2 \dots N - 1$ ako vlk.

Z toho mimochodom vyplýva, že zmeniť podobu musíš v niektorom z miest s číslami $L_i, L_i + 1, \dots, R_i$.

Pre každý výlet ťa teraz zaujíma, či je ho vôbec možné absolvovať. Teda či existuje nejaká (ľubovoľne dlhá) postupnosť ľubovoľne veľa ciest a presne jednej zmeny podoby, ktorá ťa dostane z mesta S_i do mesta E_i spĺňajúc všetky vyššie uvedené podmienky.

Implementačné detaily

Tvojou úlohou je naprogramovať nasledujúcu funkciu:

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

- N : počet miest Slovenska.
- X a Y : polia dĺžky M popisujúce cesty. Pre všetky j ($0 \leq j \leq M - 1$) platí, že mesto $X[j]$ je spojené obojsmernou cestou s mestom $Y[j]$.
- S , E , L a R : polia dĺžky Q popisujúce jednotlivé výlety - začiatok, koniec a ohraničenia L_i a R_i .

Všimni si, že hodnoty M a Q nie sú zadané explicitne. Vieš ich zistiť spôsobom, ktorý je popísaný v dokumente nazvanom "Implementation Notice".

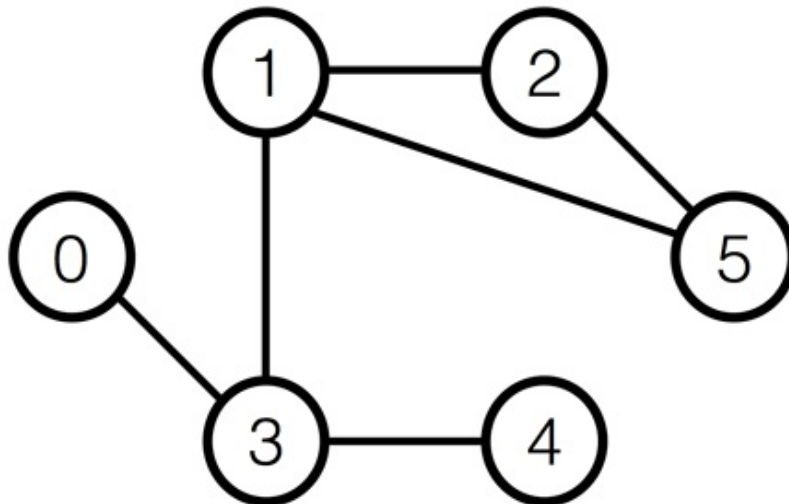
Funkciu `check_validity` grader zavolá práve raz pre každý test case. Výstupom tejto funkcie by malo byť pole A dĺžky Q . Hodnota $A[i]$ (pre $0 \leq i \leq Q - 1$) má byť:

- 1, ak sa vieš dostať dovoleným spôsobom z mesta S_i do mesta E_i
- 0, ak sa tam dostať nevieš.

Príklad

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

Grader zavolá funkciu `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])`.



V nultom výlete je možné prejsť z mesta 4 do mesta 2 nasledovným spôsobom:

- Začni v meste 4 ako človek.
- Presuň sa do mesta 3, stále ako človek.
- Presuň sa do mesta 1, stále ako človek.
- Zmeň sa na vlka v meste číslo 1.

- Presuň sa do mesta 2 ako vlk.
- Ukonči výlet.

Pre výlety 1 a 2 nie je možné absolvovať požadovanú cestu, tvoj program by mal preto vrátiť pole $[1, 0, 0]$.

Súbory `sample-01-in.txt` a `sample-01-out.txt`, ktoré máte k dispozícii v priloženom zip súbore, zodpovedajú tomuto príkladu. V zip súbore sú okrem toho aj ďalšie príklady vstupu a výstupu.

Obmedzenia

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- Pre každé j od 0 po $M - 1$ vrátane:
 - $0 \leq X_j \leq N - 1$
 - $0 \leq Y_j \leq N - 1$
 - $X_j \neq Y_j$
- Medzi každou dvojicou miest vedie nejaká postupnosť ciest, ktorá ich spája.
- Každá dvojica miest je spojená nanajviš jednou priamou cestou. Formálne, pre všetky j, k také, že $0 \leq j < k \leq M - 1$, platí $(X_j, Y_j) \neq (X_k, Y_k)$ a $(X_j, Y_j) \neq (Y_k, X_k)$.
- Pre každé i od 0 po $Q - 1$ vrátane:
 - $0 \leq L_i \leq S_i \leq N - 1$
 - $0 \leq E_i \leq R_i \leq N - 1$
 - $S_i \neq E_i$
 - $L_i \leq R_i$

Podúlohy

1. (7 bodov) $N \leq 100, M \leq 200, Q \leq 100$
2. (8 bodov) $N \leq 3\,000, M \leq 6\,000, Q \leq 3\,000$
3. (34 bodov) $M = N - 1$ a z každého mesta vedú nanajviš dve cesty (mestá teda tvoria jednu súvislú čiaru)
4. (51 bodov) bez ďalších obmedzení

Sample grader

Sample grader číta vstup v nasledujúcom formáte:

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

Sample grader vypíše výstup funkcie `check_validity` v nasledujúcom formáte:

- riadok $1 + i$ ($0 \leq i \leq Q - 1$): `A[i]`