



Vlkodlak

V prefektuře Ibaraki je N měst a M cest. Města jsou očíslována od 0 do $N - 1$ v rostoucím pořadí dle počtu jejich obyvatel. Každá cesta spojuje dvě různá města a lze ji používat v obou směrech. Z každého města se dá po cestách dojít do každého jiného města.

Máte plánováno Q výletů očíslovaných od 0 do $Q - 1$. V rámci výletu i (kde $0 \leq i \leq Q - 1$) chcete dojít z města S_i do města E_i .

Problém je, že jste vlkodlaci; na začátku každé cesty vypadáte jako **lidé**, ale během ní se musíte (právě jednou) **proměnit** ve **vlka**. K této proměně musí dojít v jednom z měst, ale můžete si určit ve kterém (může to být i město S_i nebo E_i).

Život vlkodlaka je těžký. V lidské formě se musíte vyhýbat městům s málo obyvateli a jako vlci naopak městům s mnoha obyvateli. Přesněji, pro výlet i máte dána čísla L_i a R_i (kde $0 \leq L_i \leq R_i \leq N - 1$), v lidské formě nesmíte vstoupit do měst $0, 1, \dots, L_i - 1$ a jako vlci nesmíte vstoupit do měst $R_i + 1, R_i + 2, \dots, N - 1$. Z toho plyne, že proměnit se můžete pouze v jednom z měst $L_i, L_i + 1, \dots, R_i$.

Pro každý výlet rozhodněte, zda ho lze v rámci popsaných omezení uskutečnit. Není žádné omezení na to, kolik cest můžete v rámci výletu projít.

Implementační detaily

Vaším úkolem je implementovat následující funkci:

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

- N : počet měst.
- X a Y : pole délky M . Pro každé j ($0 \leq j \leq M - 1$) je město $X[j]$ spojeno přímou cestou s městem $Y[j]$.
- S , E , L a R : pole délky Q popisující výlety.

Hodnoty M a Q lze získat jako délky polí X a S způsobem popsaným na papíru *Poznámky k implementaci (den 1)*, viz tabulka v sekci Typy.

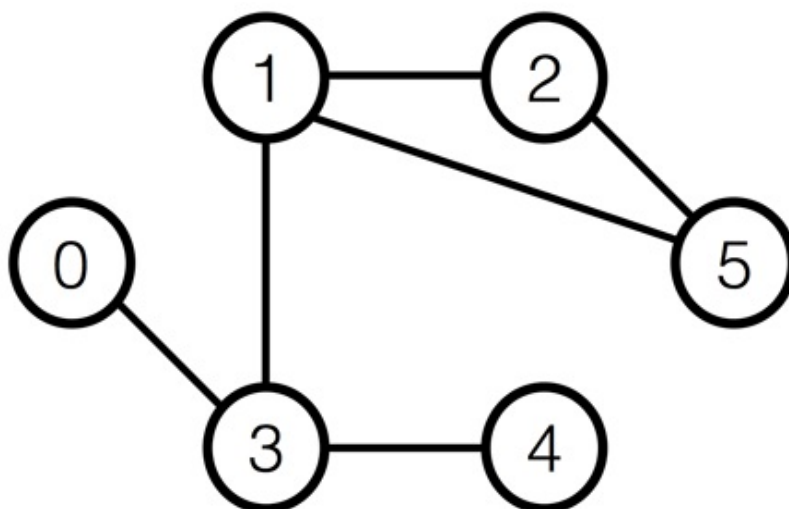
Funkce `check_validity` je volána právě jednou pro každý testovací vstup. Tato funkce má vrátit pole A délky Q obsahující celá čísla. Hodnota A_i ($0 \leq i \leq Q - 1$) má být 1,

pokud je možno podniknout výlet z města S_i do města E_i tak, že dokud jste v lidské formě, nebudete navštěvovat města $0, 1, \dots, L_i - 1$, zatímco jako vlci nebudete navštěvovat města $R_i + 1, R_i + 2, \dots, N - 1$. Jestliže takový výlet nelze podniknout, hodnota A_i má být 0.

Příklad

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

Vyhodnocovač volá `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])`.



Pro výlet 0 můžete z města 4 do města 2 cestovat následujícím způsobem:

- Začnete ve městě 4 (jste v lidské formě)
- Půjdete do města 3 (jste v lidské formě)
- Půjdete do města 1 (jste v lidské formě)
- Proměníte se na vlka (jste vlk)
- Půjdete do města 2 (jste vlk)

Pro výlety 1 a 2 nelze mezi zadanými městy cestovat.

Váš program by tak měl vrátit `[1, 0, 0]`.

Soubory `sample-01-in.txt` a `sample-01-out.txt` v zazipovaném archívu odpovídají tomuto příkladu. V archívu také najdete jeden další testovací vstup a výstup.

Omezení

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$

- Pro všechna celá čísla j splňující $0 \leq j \leq M - 1$:
 - $0 \leq X_j \leq N - 1$
 - $0 \leq Y_j \leq N - 1$
 - $X_j \neq Y_j$
- Z každého města se dá pomocí cest dostat do každého jiného města.
- Mezi každými dvěma městy vede nejvýše jedna přímá cesta. Jinak řečeno, pro všechna j a k taková, že $0 \leq j < k \leq M - 1$, platí $(X_j, Y_j) \neq (X_k, Y_k)$ a $(Y_j, X_j) \neq (X_k, Y_k)$.
- Pro všechna celá čísla i splňující $0 \leq i \leq Q - 1$:
 - $0 \leq L_i \leq S_i \leq N - 1$
 - $0 \leq E_i \leq R_i \leq N - 1$
 - $S_i \neq E_i$
 - $L_i \leq R_i$

Podúlohy

1. (7 bodů) $N \leq 100$, $M \leq 200$, $Q \leq 100$
2. (8 bodů) $N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$
3. (34 bodů) $M = N - 1$ a žádné město není propojeno přímou cestou s více než 2 jinými městy (města jsou tedy cestami propojena tak, že tvoří řadu)
4. (51 bodů) Žádná omezení navíc

Ukázkový testovač

Ukázkový testovač čte vstup v následujícím formátu:

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

Ukázkový testovač vytiskne návratovou hodnotu funkce `check_validity` v následujícím formátu:

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