



Врколак

Во областа Ибараки постојат N градови и M патишта. Градовите се нумерирани со целите броеви од 0 до $N - 1$, во растечки редослед според нивната популација. Секој пат директно поврзува еден пар од различни градови, и по секој пат може да се патува во двете насоки. Може да патувате од кој било град до кој било друг град користејќи еден или повеќе од овие патишта.

Вие сте испланирале Q патувања, нумерирани со целите броеви од 0 до $Q - 1$. Патувањето i ($0 \leq i \leq Q - 1$) е патување од градот S_i до градот E_i .

Вие сте врколак. Имате два облика: **човечки облик** и **облик на волк**. На почетокот од секое патување вие сте во човечкиот облик. На крајот од секое патување, мора да бидете во обликот на волк. За време на патувањето вие мора да се **трансформирате** (промените од човечки облик во облик на волк) точно еднаш. Можете да се трансформирате само тогаш кога се наоѓате во некој град (може да биде и градот S_i или E_i).

Да се живее како врколак не е лесно. Вие морате да ги одбегнувате ретко населените градови тогаш кога имате човечки облик, и да ги одбегнувате густо населените градови тогаш кога имате облик на волк. За секое патување i ($0 \leq i \leq Q - 1$), постојат две прагови вредности L_i и R_i ($0 \leq L_i \leq R_i \leq N - 1$) кои означуваат кои градови мора да се одбегнат. Поконкретно, морате да ги одбегнете градовите $0, 1, \dots, L_i - 1$ тогаш кога имате човечки облик, и морате да ги одбегнете градовите $R_i + 1, R_i + 2, \dots, N - 1$ тогаш кога имате облик на волк. Ова значи дека во патувањето i , можете да се трансформирате само во еден од градовите $L_i, L_i + 1, \dots, R_i$.

Ваша задача е, за секое патување, да одредите дали е возможно да се најде рута за патувањето од градот S_i до градот E_i што ќе ги задоволува погоре наведените ограничувања. Рутата на патувањето (низа од изминати директни патишта помеѓу парови од градови која започнува со S_i и завршува со E_i) може да има произволна должина.

Имплементациски детали

Треба да ја имплементирате следната функција:

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

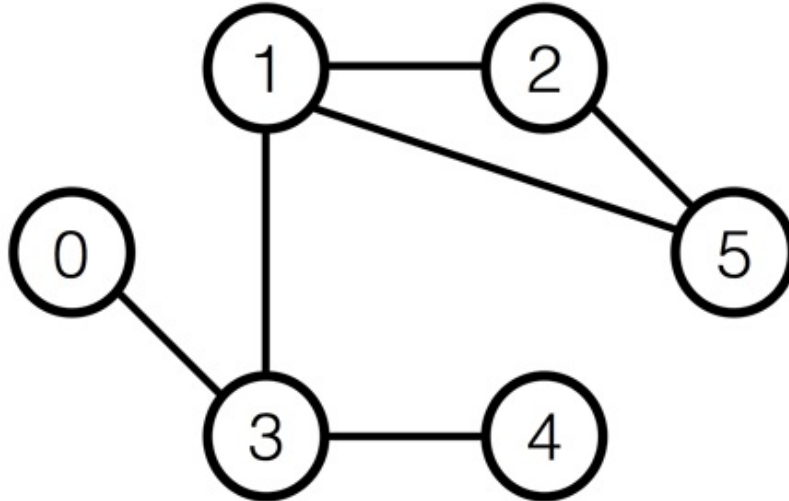
- N : бројот на градови.
- X и Y : низа со должина M . За секое j ($0 \leq j \leq M - 1$), градот $X[j]$ е директно поврзан со градот $Y[j]$ преку пат.
- S , E , L и R : низи со должина Q , што ги претставуваат патувањата.

Да забележиме дека вредностите M и Q се должините на низите, и може да се добијат како што е објаснето во Забелешки за имплементацијата (Ден 1).

Функцијата `check_validity` се повикува точно еднаш за секој тест случај. Оваа функција треба да врати низа A од цели броеви со должина Q . Вредноста на A_i ($0 \leq i \leq Q - 1$) мора да биде 1 ако е возможно да се најде рута за патувањето i што ќе ги задоволува погоре наведените услови, односно да биде 0 во спротивно.

Пример

Нека $N = 6$, $M = 6$, $Q = 3$, $X = [5, 1, 1, 3, 3, 5]$, $Y = [1, 2, 3, 4, 0, 2]$, $S = [4, 4, 5]$, $E = [2, 2, 4]$, $L = [1, 2, 3]$ и $R = [2, 2, 4]$. Оценувачот ја повикува `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])`.



За патувањето 0, рута од градот 4 до градот 2 би била следнава:

- Почнете од градот 4 (Вие имате човечки облик)
- Придвижете се до градот 3 (Вие имате човечки облик)
- Придвижете се до градот 1 (Вие имате човечки облик)
- Трансформирајте се во облик на волк (Вие имате облик на волк)
- Придвижете се до градот 2 (Вие имате облик на волк)

За патувањата 1 и 2, не можете да најдете рута помеѓу дадените градови што ќе ги

задоволува наведените услови. Според тоа, вашата програма треба да врати $[1, 0, 0]$.

Датотеките `sample-01-in.txt` и `sample-01-out.txt` во zip архивата одговараат на овој пример. Оваа архива исто така содржи и друг пар од примери за влезна/излезна датотека.

Ограничувања

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- За секое $0 \leq j \leq M - 1$
 - $0 \leq X_j \leq N - 1; 0 \leq Y_j \leq N - 1$
 - $X_j \neq Y_j$
- Може да патувате од кој било град до кој било друг град користејќи ги патиштата.
- Секој пар од градови се директно поврзани најмногу со еден пат. Со други зборови, за секое $0 \leq j < k \leq M - 1$, $(X_j, Y_j) \neq (X_k, Y_k)$ и $(Y_j, X_j) \neq (X_k, Y_k)$.
- За секое $0 \leq i \leq Q - 1$
 - $0 \leq L_i \leq S_i \leq N - 1$
 - $0 \leq E_i \leq R_i \leq N - 1$
 - $S_i \neq E_i; L_i \leq R_i$

Подзадачи

1. (7 поени) $N \leq 100, M \leq 200, Q \leq 100$
2. (8 поени) $N \leq 3\,000, M \leq 6\,000, Q \leq 3\,000$
3. (34 поени) $M = N - 1$ и од секој град поаѓаат најмногу 2 пата (градовите се поврзани во една линија)
4. (51 поени) Нема дополнителни ограничувања

Пример оценувач

Пример оценувачот го чита влезот во следниот формат:

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

Пример оценувачот ја печати вредноста што ја враќа функцијата `check_validity` во следниот формат:

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