



Вовкулака

У префектурі Ібаракі в Японії є N міст та M доріг. Міста пронумеровано від 0 до $N - 1$ в порядку зростання їх населення. Кожна дорога з'єднує пару різних міст, по ній можна рухатись в обох напрямках. Ви можете подорожувати з кожного міста до кожного іншого використовуючи одну чи більше з цих доріг.

Ви запланували Q подорожей, пронумерованих від 0 до $Q - 1$. Подорож i ($0 \leq i \leq Q - 1$) починається в місті S_i і закінчується в місті E_i .

Ви - вовкулака. Ви можете знаходитись у двох формах: у **формі людини** і у **формі вовка**. На початку кожної подорожі ви знаходитесь у формі людини. В кінці кожної подорожі ви маєте бути у формі вовка. Протягом подорожі ви маєте **перетворитися** (змінити людську форму на вовчу) тільки один раз і це має відбутись коли ви знаходитесь у якомусь місті (можливо S_i або E_i).

Бути вовкулакою не легко. Ви маєте оминати малонаселені міста коли ви маєте людську форму та густонаселені - коли ви маєте форму вовка. Для кожної подорожі i є два порогових значення L_i та R_i ($0 \leq L_i \leq R_i \leq N - 1$), що показують, які міста треба оминати. Більш точно, ви маєте оминати міста $0, 1, \dots, L_i - 1$ коли ви маєте форму людини, та міста $R_i + 1, R_i + 2, \dots, N - 1$ коли ви у формі вовка. Це означає, що у подорожі i ви можете перетворюватись тільки в одному з міст $L_i, L_i + 1, \dots, R_i$.

Для кожної подорожі вам потрібно визначити чи можна проїхати з міста S_i до міста E_i , у спосіб, що задовольняє неведені обмеження. Маршрут, по якому ви рухаєтесь, може бути довільної довжини.

Деталі реалізації

Ви маєте реалізувати наступну функцію:

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

- N : кількість міст.
- X та Y : масиви довжини M . Для кожного j ($0 \leq j \leq M - 1$), місто $X[j]$ безпосередньо з'єднано дорогою з містом $Y[j]$.
- S , E , L , та R : масиви довжини Q , що задають подорожі.

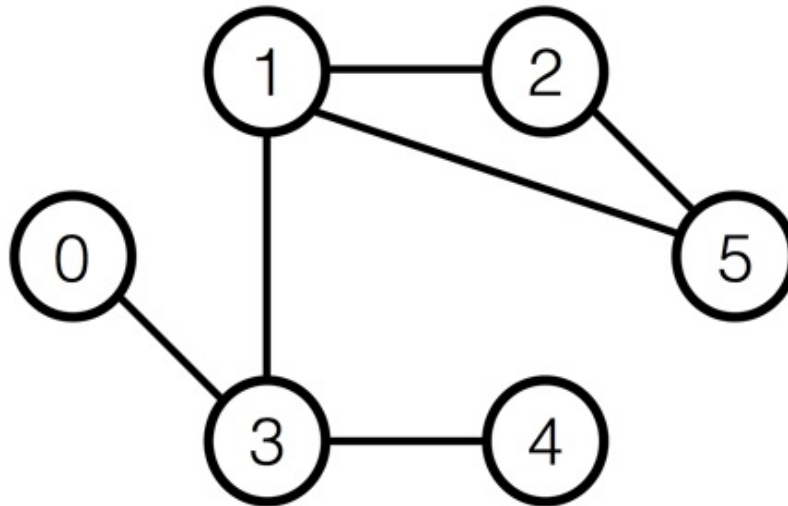
Зауважте, що значення M та Q є довжинами масивів та їх можна отримати як зазначено в зауваженнях до реалізації.

Функція `check_validity` викликається один раз для кожного теста. Ця функція має повертати масив цілих чисел A довжини Q . Значення A_i ($0 \leq i \leq Q - 1$) має бути 1 якщо подорож i є можливою з виконанням перелічених умов та 0 в протилежному випадку.

Приклад

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

Модуль перевірки викликає `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])`.



Для подорожі 0, ви можете проїхати з міста 4 до міста 2 наступним чином:

- Почати в місті 4 (Ви у формі людини)
- Пересунутись у місто 3 (Ви у формі людини)
- Пересунутись у місто 1 (Ви у формі людини)
- Перетворитись у форму вовка (Ви у формі вовка)
- Пересунутись у місто 2 (Ви у формі вовка)

Для подорожей 1 та 2, немає можливості проїзду між заданими містами.

Отже, ваша функція має повернути $[1, 0, 0]$.

Файли `sample-01-in.txt` та `sample-01-out.txt`, що знаходяться у zip-архіві відповідають прикладу. Там також є ще одна пара прикладів вводу-виводу.

Обмеження

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- Для кожного $0 \leq j \leq M - 1$
 - $0 \leq X_j \leq N - 1$
 - $0 \leq Y_j \leq N - 1$
 - $X_j \neq Y_j$
- По дорогах ви можете потрапити з довільного міста до якого завгодно іншого міста.
- Кожна пара міст безпосередньо зеднана не більш ніж однією дорогою. Іншими словами для усіх $0 \leq j < k \leq M - 1$, $(X_j, Y_j) \neq (X_k, Y_k)$ та $(Y_j, X_j) \neq (X_k, Y_k)$.
- Для кожного $0 \leq i \leq Q - 1$
 - $0 \leq L_i \leq S_i \leq N - 1$
 - $0 \leq E_i \leq R_i \leq N - 1$
 - $S_i \neq E_i$
 - $L_i \leq R_i$

Підзадачі

1. (7 балів) $N \leq 100$, $M \leq 200$, $Q \leq 100$
2. (8 балів) $N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$
3. (34 бали) $M = N - 1$ і кожне з міст безпосередньо з'єднано з не більше ніж 2 містами (міста з'єднані у лінію)
4. (51 бал) Без додаткових обмежень.

Модуль перевірки з прикладу

Модуль перевірки з прикладу читає дані у такому форматі:

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

Модуль перевірки з прикладу друкує значення, що повертає `check_validity` у наступному форматі:

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