



Върколак

В японската префектура Ибараки има N града и M свързващи ги шосета. Градовете са номерирани от 0 до $N - 1$ в нарастващ ред според броя на жителите им. Всяко шосе свързва два различни града и по него може да се минава и в двете посоки. От всеки град до всеки град съществува път, който преминава по наличните шосета. Планирали сте Q пътувания, номерирани от 0 до $Q - 1$. Пътуване с номер i ($0 \leq i \leq Q - 1$) е от града с номер S_i до града с номер E_i .

Вие сте върколак, който може да се намира в две форми – **човешка** или **вълча**. В началото на всяко пътуване се намирате в човешката си форма, а в края му трябва да сте във вълчата. По време на пътуването трябва да се **преобразите** от едната си форма (човешката) в другата (вълчата) точно един път в който и да е от градовете на пътуването (включително S_i или E_i).

Животът на върколака не е лесен. От опит знаете, че при пътуване е добре да избягвате слабо населените градове, когато сте в човешка форма и многолюдните градове, когато сте във вълча. По-точно, за всяко пътуване i избирате две числа L_i и R_i такива, че $0 \leq L_i \leq R_i \leq N - 1$ и по време на пътуването избягвате градовете $0, 1, \dots, L_i - 1$, когато сте в човешка форма и градовете $R_i + 1, R_i + 2, \dots, N - 1$, когато сте във вълча. Т.е. смяната от едната форма в другата трябва да стане в някой от градовете $L_i, L_i + 1, \dots, R_i$.

Задачата Ви е, за всяко от пътуванията да определите дали пътуването е възможно при така зададените условия. Пътят който ще изберете може да има произволна дължина.

Детайли за реализацията

Напишете следната функция:

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

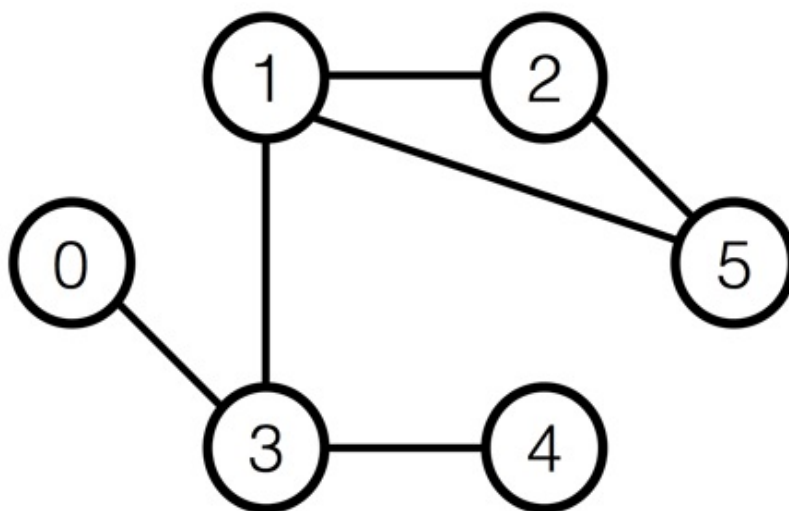
- N : броят градове.
- X и Y : масиви с дължина M . За всяко j ($0 \leq j \leq M - 1$), град $X[j]$ е свързан с шосе с град $Y[j]$.
- S , E , L , и R : масиви с дължина Q , описващи зададените пътувания.

Функцията `check_validity` ще бъде извикана точно веднъж за всеки тест. Функцията трябва да връща масив от цели числа A с дължина Q . Стойността на A_i ($0 \leq i \leq Q - 1$) трябва да бъде равна на 1 ако е възможно да се пътува от град S_i до град E_i , избягвайки градове $0, 1, \dots, L_i - 1$ в човешка форма и избягвайки градове $R_i + 1, R_i + 2, \dots, N - 1$ във вълча форма. В противен случай стойността на A_i трябва да е равна на 0.

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 $R = [2, 2, 4]$.

Грейдърът ще извика `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])`.



Пътуване с номер 0 от град 4 до град 2 може да бъде извършено по следния начин:

- Започвате от град 4 (Намирате се в човешка форма)
- Пътувате до град 3 (Намирате се в човешка форма)
- Пътувате до град 1 (Намирате се в човешка форма)
- Трансформирате се във вълча форма (Вече се намирате във вълча форма)
- Пътувате до град 2 (Намирате се във вълча форма)

Пътувания с номера 1 и 2 не могат да бъдат осъществени, т.е. вашата функция трябва да върне `[1, 0, 0]`.

Файловете `sample-01-in.txt` и `sample-01-out.txt` в прикачения zip архив съответстват на описания пример. В архива има и други тестови примери.

Ограничения

- $2 \leq N \leq 200\,000$

- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- За всяко $0 \leq j \leq M - 1$
 - $0 \leq X_j \leq N - 1$
 - $0 \leq Y_j \leq N - 1$
 - $X_j \neq Y_j$
- Съществува път между всеки два града, преминаващ по зададените шосета.
- Всяка двойка градове е свързана с най-много едно шосе. С други думи - за всеки $0 \leq j < k \leq M - 1$, $(X_j, Y_j) \neq (X_k, Y_k)$ and $(Y_j, X_j) \neq (X_k, Y_k)$.
- За всяко $0 \leq i \leq Q - 1$
 - $0 \leq L_i \leq S_i \leq N - 1$
 - $0 \leq E_i \leq R_i \leq N - 1$
 - $S_i \neq E_i$
 - $L_i \leq R_i$

Подзадачи

1. (7 точки) $N \leq 100$, $M \leq 200$, $Q \leq 100$
2. (8 точки) $N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$
3. (34 точки) $M = N - 1$ и всеки град е свързан с най-много 2 града (градовете са свързани в "линия")
4. (51 точки) Няма допълнителни ограничения

Примерен грейдър

Примерният грейдър чете входните данни в следния формат:

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

Примерният грейдър извежда върнатия от функцията `check_validity` в следния формат:

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