Оборотень

В префектуре Ибараки в Японии N городов и M дорог. Города пронумерованы от 0 до $N-1$ по возрастанию их численности. Каждая дорога соединяет пару различных городов, и по ней можно перемещаться в обоих направлениях. Из любого города в любой другой можно добраться, переместившись по одной или нескольким дорогам.

Вы запланировали Q путешествий, пронумерованных от 0 до $Q-1$. Путешествие i ($0 \leq i \leq Q-1$) должно начаться в городе S_i и закончиться в городе E_i.

Вы — оборотень. Вы можете принимать две формы: **форму человека** и **форму волка**. В начале каждого путешествия вы находитесь в форме человека, а в конце каждого путешествия вы должны быть в форме волка. Во время путешествия вам необходимо **превратиться** (сменить форму человека на форму волка) ровно один раз. Вы можете превращаться, только находясь в каком-либо городе (возможно, в городе S_i или E_i).

Жизнь оборотня непроста. В форме человека вам необходимо избегать малонаселённых городов, а в форме волка — густонаселённых городов. Для каждого путешествия i ($0 \leq i \leq Q-1$) заданы два числа L_i и R_i ($0 \leq L_i \leq R_i \leq N-1$), которые описывают, какие города вам необходимо избегать. Более конкретно, необходимо избегать города $0, 1, \ldots, L_i-1$, находясь в форме человека, а также города $R_i+1, R_i+2, \ldots, N-1$, находясь в форме волка. В частности, это означает, что вам необходимо превратиться в одном из городов $L_i, L_i+1, \ldots, R_i$.

Ваша задача — для каждого путешествия определить, возможно ли добраться из города S_i в город E_i, чтобы описанные выше ограничения выполнялись. Ваш путь может иметь произвольную длину.

Детали реализации

Вам требуется реализовать следующую функцию:

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

- `N`: количество городов.

- `X` и `Y`: массивы длины M. Для каждого j ($0 \leq j \leq M-1$), города `X[j]` и `Y[j]` напрямую связаны дорогой.
- `S`, `E`, `L` и `R`: массивы длины Q, описывающие путешествия.

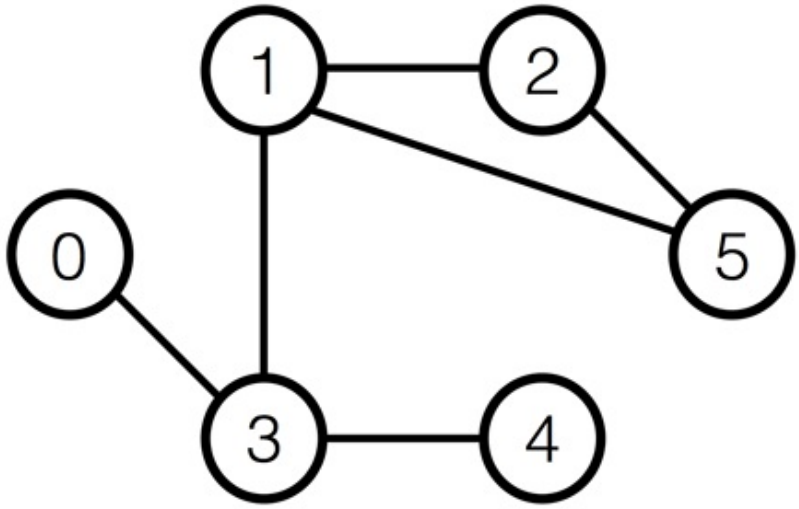
Обратите внимание, что значения M и Q — это длины массивов, способ получения которых описан в памятке о деталях реализации.

Функция `check_validity` вызывается ровно один раз для каждого теста. Функция должна вернуть массив A из Q целых чисел. Значение A_i ($0 \leq i \leq Q-1$) должно быть равно `1`, если путешествие i можно совершить согласно описанным ограничениям, и `0` в противном случае.

Пример

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

Проверяющий модуль (grader) совершит вызов `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])`.



В путешествии 0 можно добраться из города 4 в город 2 следующим образом:

- Начать в городе 4 (вы в форме человека)
- Переместиться в город 3 (вы в форме человека)
- Переместиться в город 1 (вы в форме человека)
- Превратиться в волка (вы в форме волка)
- Переместиться в город 2 (вы в форме волка)

В путешествиях 1 и 2 между данными парами городов переместиться невозможно.

Таким образом, ваша программа должна вернуть $[1, 0, 0]$.

Файлы `sample-01-in.txt` и `sample-01-out.txt` в приложенном архиве соответствуют этому примеру. В архиве также находятся другая пара файлов

ввода и вывода, соответствующих еще одному примеру.

Ограничения

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- Для каждого $0 \leq j \leq M - 1$:
  - $0 \leq X_j \leq N - 1$
  - $0 \leq Y_j \leq N - 1$
  - $X_j \neq Y_j$
- От любого города до любого другого можно добраться по дорогам.
- Каждая пара городов напрямую связана не более, чем одной дорогой. Другими словами, для всех пар индексов $0 \leq j < k \leq M - 1$ верно $(X_j, Y_j) \neq (X_k, Y_k)$ и $(Y_j, X_j) \neq (X_k, Y_k)$.
- Для каждого $0 \leq i \leq Q - 1$:
  - $0 \leq L_i \leq S_i \leq N - 1$
  - $0 \leq E_i \leq R_i \leq N - 1$
  - $S_i \neq E_i$
  - $L_i \leq R_i$

Подзадачи

1. (7 баллов) $N \leq 100$, $M \leq 200$, $Q \leq 100$
2. (8 баллов) $N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$
3. (34 балла) $M = N - 1$, а также каждый город является концом не более двух дорог (города соединены в цепочку)
4. (51 балл) Нет дополнительных ограничений

Пример проверяющего модуля

Пример проверяющего модуля считывает ввод в следующем формате:

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

Пример проверяющего модуля выводит значение, возвращаемое функцией `check_validity`, в следующем формате:

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