狼男

茨城県には N 個の都市と M 本の道路が存在する. 都市には人口の少ない順に 0 から $N-1$ までの番号がつけられている. それぞれの道路は異なる 2 つの都市をつないでいて, 双方向に移動することができる. どの 2 つの異なる都市の間もいくつかの道路を通って移動することができる.

あなたは 0 から $Q-1$ までの番号がつけられた Q 個の旅行を計画している. i 番目 $(0 \leqq i \leqq Q-1)$ の旅行を旅行 i と呼ぶ. 旅行 i では都市 S_i から都市 E_i へ移動する.

あなたは狼男であり, **人間の姿**と**狼の姿**の 2 つの姿をもつ. それぞれの旅行の開始時には, あなたは人間の姿である. それぞれの旅行の終了時には, あなたは狼の姿でなければならない. 旅行の間にあなたはちょうど 1 回**変身**しなければならない. ここで, 変身は人間の姿から狼の姿になることを指す. 変身は, あなたがいずれかの都市にいる間にのみすることができる(その都市は S_i または E_i でもよい).

狼男として生きることは簡単ではない. 人間の姿のでいるときは人口が少ない都市を避けなければならない. また, 狼の姿でいるときは人口が多い都市を避けなければならない. 各旅行 i $(0 \leqq i \leqq Q-1)$ に対し, 避けなければならない都市の基準となる 2 つのしきい値 L_i と R_i $(0 \leqq L_i \leqq R_i \leqq N-1)$ があり, 人間の姿のとき都市 $0, 1, \ldots, L_i-1$ を避け, 狼の姿のとき都市 $R_i+1, R_i+2, \ldots, N-1$ を避けなければならない. これを踏まえると, 旅行 i では都市 $L_i, L_i+1, \ldots, R_i$ のいずれかでしか変身できないことになる.

あなたの仕事は, 前述の条件を満たすように都市 S_i から都市 E_i へ移動することができるかどうかを判定することである. 移動する経路の長さに制約はない.

実装の詳細

あなたは, 次の関数を実装する必要がある:

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

- N: 都市の数である.
- X, Y: 長さ M の配列である. 各 j $(0 \leqq j \leqq M-1)$ に対し, 都市 X[j] は都市 Y[j] と 1 本の道路によって直接つながれている.
- S, E, L, R: 長さ Q の配列であり, 各旅行の情報を表す.

なお, ここで M と Q は配列の長さを表す値であり, これらは「実装上の注意」に記されている方法で取得することができる.

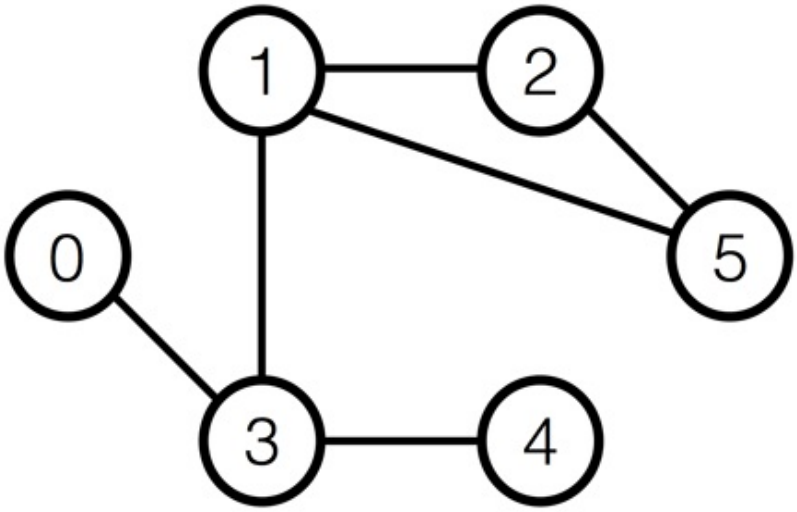
関数 check_validity はそれぞれのテストケースに対し, ちょうど 1 回呼び出される. この関数は長さ Q

の整数の配列 A を返さなければならない. 各 i ($0 \leqq i \leqq Q-1$) に対し, A_i は, 旅行 i が問題文で与えられている条件を満たすとき 1, それ以外のとき 0 でなければならない.

入出力例

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

採点プログラムは `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])` を呼び出す.



旅行 0 では, 以下の手順で都市 4 から都市 2 に移動することができる.

- 都市 4 を出発する (あなたは人間の姿である).
- 都市 3 に行く (あなたは人間の姿である).
- 都市 1 に行く (あなたは人間の姿である).
- 狼の姿に変身する (あなたは狼の姿である).
- 都市 2 に行く (あなたは狼の姿である).

旅行 1 と 2 では, 与えられた都市の間を条件に沿って移動することができない.

ゆえに, あなたのプログラムは $[1, 0, 0]$ を返さなければならない.

zip 圧縮された添付パッケージ (attachment package) に入ったファイル `sample-01-in.txt` および `sample-01-out.txt` がこの例に対応している. 他の入出力例もこのパッケージで得られる.

制約

- $2 \leqq N \leqq 200\,000$.
- $N - 1 \leqq M \leqq 400\,000$.
- $1 \leqq Q \leqq 200\,000$.
- 各 $0 \leqq j \leqq M - 1$ に対し,
  - $0 \leqq X_j \leqq N - 1$.
  - $0 \leqq Y_j \leqq N - 1$.
  - $X_j \neq Y_j$.
- どの 2 つの異なる都市の間もいくつかの道路を通って移動することができる.
- 各都市の組はたかだか 1 本の道路によってつながっている. すなわち, 各 $0 \leqq j < k \leqq M - 1$ に対し, $(X_j, Y_j) \neq (X_k, Y_k)$ かつ $(Y_j, X_j) \neq (X_k, Y_k)$ が満たされる.
- 各 $0 \leqq i \leqq Q - 1$ に対し,
  - $0 \leqq L_i \leqq S_i \leqq N - 1$.
  - $0 \leqq E_i \leqq R_i \leqq N - 1$.
  - $S_i \neq E_i$.
  - $L_i \leqq R_i$.

小課題

1. (7 点) $N \leqq 100$, $M \leqq 200$, $Q \leqq 100$.
2. (8 点) $N \leqq 3\,000$, $M \leqq 6\,000$, $Q \leqq 3\,000$.
3. (34 点) $M = N - 1$ であり, 各都市はたかだか 2 本の道路と接続している (都市は直線状につながっている).
4. (51 点) 追加の制約はない.

採点プログラムのサンプル

採点プログラムのサンプルの入力形式は以下の通りである.

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

採点プログラムのサンプルは以下の形式で `check_validity` の返り値を出力する.

- $1 + i$ 行目 ($0 \leqq i \leqq Q - 1$): A_i.