



## Werewolf (人狼)

在日本的茨縣內共有  $N$  個城市及  $M$  條道路。這些城市是根據它的人口升序的順序編號為  $0$  至  $N - 1$ 。每條道則把兩個不同的城市連接起來，並且可以雙向行走。你可以從任何一個城市通過這些道路去到另外任何一個城市。

你正在計劃  $Q$  個行程，這些行程分別編號為  $0$  至  $Q - 1$ 。而第  $i$  ( $0 \leq i \leq Q - 1$ ) 個行程就是要由城市  $S_i$  去到城市  $E_i$ 。

你本身實際上是一隻人狼。你有兩個型態: 就是人的型態及狼的型態。在每個行程的開始時，你是人的型態。而在行程結束時，你必須是狼的型態。在行程中，你必須要變身 (即由人的型態變成狼的型態) 正好一次，且變身時要在其中一個城市內 (包括可能在  $S_i$  或  $E_i$  內)。

人狼的生活並不容易。當你是人的型態時，你必須避開人口少的城市，而當你是狼的型態時，你必須避開人口多的城市。在每一次行程  $i$  ( $0 \leq i \leq Q - 1$ ) 中，均存在兩個界限值  $L_i$  及  $R_i$  ( $0 \leq L_i \leq R_i \leq N - 1$ ) 它們是用以表示那些城市必須要避開。準確來說，當你是人的型態時你要避開城市  $0, 1, \dots, L_i - 1$ ，而當你是狼的型態時則要避開城市  $R_i + 1, R_i + 2, \dots, N - 1$ 。換言之你在第  $i$  次行程中，你必須在城市  $L_i, L_i + 1, \dots, R_i$  中的其中一個內變身。

對每一次行程，你的工作就是要決定是否有可能由城市  $S_i$  去到城市  $E_i$  且符合上面要求的限制條件。你的行程可以有任意的長度。

## Implementation details (實現細則)

你需要實現以下的函數:

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

- $N$ : 城市的數目
- $X$  和  $Y$ : 分別為長度為  $M$  的矩陣。對於每個  $j$  ( $0 \leq j \leq M - 1$ ) 而言，城市  $X[j]$  有道路直接通往城市  $Y[j]$ 。
- $S, E, L$ , 及  $R$ : 它們均為長度為  $Q$  的矩陣，它們代表著每次行程的資料。

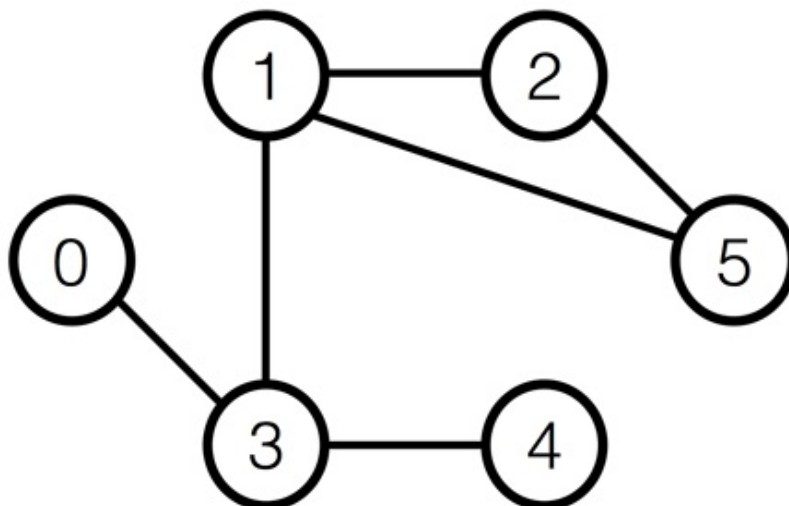
請注意  $M$  及  $Q$  為矩陣的長度，它們的值是可以通過 "編程注意事項" 內的相關說明取得。

對於每個測試數據程而言，`check_validity` 只被調用緊一次。這函數應返回一長度為  $Q$  的矩陣  $A$ 。而  $A_i$  ( $0 \leq i \leq Q - 1$ ) 的值必須是  $1$  若該行程是有可能在滿足上面要求的條件下完成的，否則其值必須是  $0$ 。

## Example (樣例)

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

樣例評分程式將以以下的方式調用 `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])`.



對於行程 0，你可以以下方式由城市 4 去可城市 2:

- 起點為城市 4 (你是人的型態)
- 前往到城市 3 (你是人的型態)
- 再前往城市 1 (你是人的型態)
- 你變身為狼 (你現在是狼的型態)
- 你前往城市 2 (你是狼的型態)

而對於行程 1 及 2，你則不可能完成在指定的城市間行程。

因此，你的程式必須返回 `[1, 0, 0]`。

在附件的壓縮檔案包有 `sample-01-in.txt` 及 `sample-01-out.txt` 這兩個檔案，它們就是對應這些樣例的。另外的一些輸入輸出例子亦可以在這檔案包內找到。

## Constraints (限制條件)

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- 對於每個  $0 \leq j \leq M - 1$ 
  - $0 \leq X_j \leq N - 1$
  - $0 \leq Y_j \leq N - 1$
  - $X_j \neq Y_j$

- 你可以通過道路由任何一個城市去到另外任何一個城市。
- 每一對城市最多只由一條道路直接連起來。換言之，對於所有  $0 \leq j < k \leq M - 1$  而言， $(X_j, Y_j) \neq (X_k, Y_k)$  且  $(Y_j, X_j) \neq (X_k, Y_k)$
- 對於  $0 \leq i \leq Q - 1$ 
  - $0 \leq L_i \leq S_i \leq N - 1$
  - $0 \leq E_i \leq R_i \leq N - 1$
  - $S_i \neq E_i$
  - $L_i \leq R_i$

## Subtasks (子任務)

1. (7 分)  $N \leq 100, M \leq 200, Q \leq 100$
2. (8 分)  $N \leq 3\,000, M \leq 6\,000, Q \leq 3\,000$
3. (34 分)  $M = N - 1$  且沒有一個城市是直接通往多於 2 個其他城市的 (所有城市是以一條直線的形式連起來)
4. (51 分) 沒有任何附加的限制

## Sample grader (樣例評分程式)

樣例評分程式將接受以下格的輸入資料:

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

樣例評分程式將用以下格式把 `check_validity` 的返回輸出:

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