



# Wilkołak

W prefekturze Ibaraki w Japonii jest  $N$  miast oraz  $M$  dróg. Miasta są ponumerowane od 0 do  $N - 1$  w rosnącej kolejności, zgodnie z ich populacją. Każda droga łączy pary różnych miast i może być pokonywana w oba kierunki. Możesz przemieścić się z dowolnego miasta do każdego innego miasta używając wyłącznie dróg.

Planujesz  $Q$  podróży ponumerowanych od 0 do  $Q - 1$ . Podczas  $i$ -tej podróży ( $0 \leq i \leq Q - 1$ ), planujesz przemieścić się z miasta  $S_i$  do miasta  $E_i$ .

Jesteś wilkołakiem. Masz dwie postacie: **postać człowieka** i **postać wilka**. Na początku każdej podróży jesteś w postaci człowieka. Na końcu każdej podróży musisz być w postaci wilka. Podczas podróży, musisz się **przemienić** (zmienić postać z człowieka na wilka) dokładnie raz i musi to nastąpić w pewnym mieście (być może  $S_i$  lub  $E_i$ ).

Życie wilkołaka nie jest proste. Z Twojego doświadczenia, wiesz że warto unikać miast o niskiej liczbie mieszkańców kiedy jesteś człowiekiem. Warto także unikać miast o wysokiej liczbie mieszkańców, kiedy jesteś wilkiem. Dokładniej, dla każdej podróży  $i$ , wybierasz dwie liczby  $L_i$  oraz  $R_i$  spełniające  $0 \leq L_i \leq R_i \leq N - 1$ . Oznacza to, że w  $i$ -tej podróży, zdecydowałeś się unikać miast  $0, 1, \dots, L_i - 1$ , kiedy jesteś w postaci człowieka oraz unikać miast  $R_i + 1, R_i + 2, \dots, N - 1$  kiedy jesteś w postaci wilka. W szczególności wynika z tego, że musisz się przemienić w jednym z miast  $L_i, L_i + 1, \dots, R_i$ .

Dla każdej podróży Twoim zadaniem jest określenie, czy jest możliwa podróż z miasta  $S_i$  do miasta  $E_i$ , tak aby spełnić powyższe warunki. Droga ta może mieć dowolną długość.

## Szczegóły implementacji

Powinieneś zaimplementować następującą funkcję:

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

- $N$ : liczba miast.
- $X$  oraz  $Y$ : tablice długości  $M$ . Dla każdego  $j$  ( $0 \leq j \leq M - 1$ ), miasto  $X[j]$  jest bezpośrednio połączone drogą z miastem  $Y[j]$ .

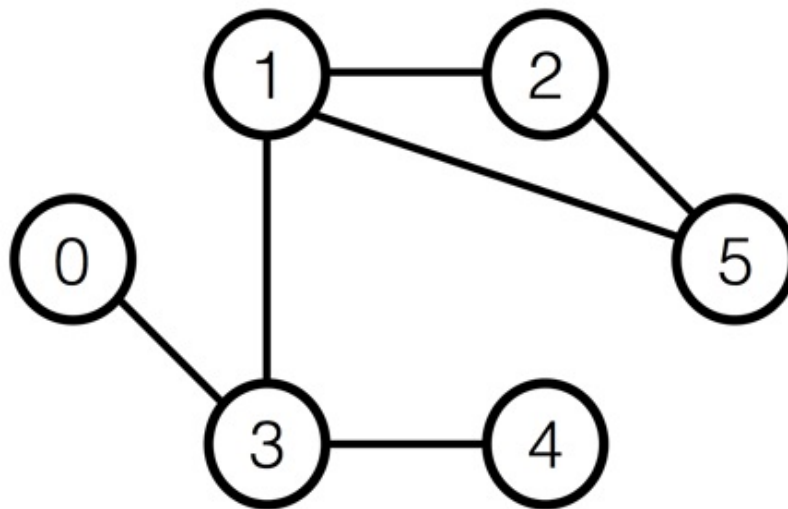
- $S$ ,  $E$ ,  $L$  oraz  $R$ : tablice o długości  $Q$ , reprezentujące podróże.

Funkcja `check_validity` jest wywoływana dokładnie raz dla każdego testu. Funkcja ta powinna zwrócić tablicę  $A$  liczb całkowitych o długości  $Q$ . Wartość  $A_i$  ( $0 \leq i \leq Q - 1$ ) ma wynosić 1, jeżeli jest możliwa podróż z miasta  $S_i$  do miasta  $E_i$ , omijająca miasta  $0, 1, \dots, L_i - 1$  kiedy jesteś w postaci człowieka oraz miasta  $R_i + 1, R_i + 2, \dots, N - 1$  kiedy jesteś w postaci wilka. W przeciwnym wypadku, wartość ta powinna równać się 0.

## Przykład

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

Sprawdzaczka wywołuje `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])`.



Dla podróży 0, możesz przemieścić się z miasta 4 do miasta 2 w następujący sposób:

- Zaczynasz w mieście 4 (Jesteś w postaci człowieka)
- Przemieszczasz się do miasta 3 (Jesteś w postaci człowieka)
- Przemieszczasz się do miasta 1 (Jesteś w postaci człowieka)
- Przemieniasz się w wilka (Jesteś w postaci wilka)
- Przemieszczasz się do miasta 2 (Jesteś w postaci wilka)

Dla podróży 1 i 2 nie możesz przemieścić się między danymi miastami zgodnie z narzuconymi warunkami.

Zatem Twój program winien zwrócić `[1, 0, 0]`.

Pliki `sample-01-in.txt` oraz `sample-01-out.txt` w załączonym archiwum odpowiadają temu przykładowi. Inne przykładowe wejścia i wyjścia są dostępne w tym archiwum.

## Ograniczenia

- $2 \leq N \leq 200\,000$
- $N - 1 \leq M \leq 400\,000$
- $1 \leq Q \leq 200\,000$
- $0 \leq X_j \leq N - 1$  ( $0 \leq j \leq M - 1$ )
- $0 \leq Y_j \leq N - 1$  ( $0 \leq j \leq M - 1$ )
- Możesz przemieścić się z dowolnego miasta do dowolnego innego miasta używając dróg.
- $X_j \neq Y_j$  ( $0 \leq j \leq M - 1$ )
- $(X_j, Y_j) \neq (X_k, Y_k)$  oraz  $(X_j, Y_j) \neq (Y_k, X_k)$  ( $0 \leq j < k \leq M - 1$ )
- $0 \leq S_i \leq N - 1$  ( $0 \leq i \leq Q - 1$ )
- $0 \leq E_i \leq N - 1$  ( $0 \leq i \leq Q - 1$ )
- $S_i \neq E_i$  ( $0 \leq i \leq Q - 1$ )
- $0 \leq L_i \leq R_i \leq N - 1$  ( $1 \leq i \leq Q - 1$ )
- $L_i \leq S_i$  ( $0 \leq i \leq Q - 1$ )
- $E_i \leq R_i$  ( $0 \leq i \leq Q - 1$ )

## Podzadania

1. (7 punktów)  $N \leq 100$ ,  $M \leq 200$ ,  $Q \leq 100$
2. (8 punktów)  $N \leq 3\,000$ ,  $M \leq 6\,000$ ,  $Q \leq 3\,000$
3. (34 punkty)  $M = N - 1$  oraz żadne miasto nie jest bezpośrednio połączone z więcej niż dwoma miastami (miasta są połączone w linię)
4. (51 punktów) Brak dodatkowych warunków.

## Przykładowa sprawdzaczka

Przykładowa sprawdzaczka wczytuje wejście w następującym formacie:

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

Przykładowa sprawdzaczka wypisuje zwróconą wartość `check_validity` w następującym formacie:

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