

# Zadanie: OSI Osiedla



XXVI OI, etap II, dzień drugi. Plik źródłowy osi.\* Dostępna pamięć: 256 MB.

14.02.2019

W Bitowicach bardzo często dochodzi do wypadków. Burmistrz miasta jest zdania, że powodem jest nadmierne skomplikowanie sieci  $m$  dwukierunkowych ulic, łączących  $n$  skrzyżowań. Postanowił on zatem rozwiązać ten problem, zmieniając każdą ulicę w ulicę jednokierunkową.

W wyniku tej akcji chciałby doprowadzić do utworzenia jak najmniejszej liczby osiedli. **Osiedlem** nazywamy każdy maksymalny (czyli niedający się rozszerzyć) zbiór skrzyżowań, taki że z dowolnego skrzyżowania z tego zbioru można przejechać do dowolnego innego skrzyżowania z tego zbioru zgodnie ze skierowaniem ulic.

Twoim zadaniem będzie napisanie programu, który znajdzie zarówno minimalną liczbę osiedli, jak i dające ją skierowanie ulic.

## Wejście

Pierwszy wiersz standardowego wejścia zawiera dwie liczby całkowite  $n$  i  $m$  ( $2 \leq n \leq 1\,000\,000$ ,  $1 \leq m \leq 1\,000\,000$ ) oddzielone pojedynczym odstępem, oznaczające odpowiednio liczbę skrzyżowań oraz liczbę ulic w Bitowicach. Skrzyżowania numerujemy liczbami od 1 do  $n$ .

Kolejne  $m$  wierszy opisuje ulice;  $i$ -ty z nich zawiera dwie liczby całkowite  $a_i, b_i$  ( $1 \leq a_i, b_i \leq n$ ,  $a_i \neq b_i$ ) oddzielone pojedynczym odstępem, oznaczające, że  $i$ -ta ulica łączy skrzyżowania o numerach  $a_i$  oraz  $b_i$ . Może istnieć więcej niż jedna ulica łącząca daną parę skrzyżowań.

## Wyjście

W pierwszym wierszu standardowego wyjścia należy wypisać jedną liczbę całkowitą oznaczającą liczbę osiedli przy optymalnym skierowaniu ulic. Drugi wiersz powinien zawierać słowo długości  $m$ , reprezentujące optymalne skierowanie;  $i$ -ty znak tego słowa określa skierowanie  $i$ -tej ulicy, zgodnie z kolejnością na wejściu. Znak  $>$  oznacza skierowanie od skrzyżowania  $a_i$  do skrzyżowania  $b_i$ , natomiast  $<$  oznacza skierowanie od skrzyżowania  $b_i$  do skrzyżowania  $a_i$ . Inne znaki nie są dozwolone. Jeżeli istnieje więcej niż jedno optymalne skierowanie, Twój program powinien wypisać dowolne z nich.

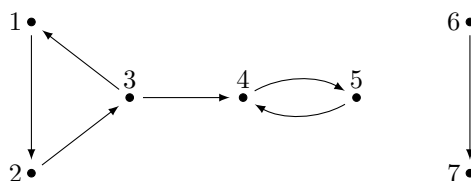
## Przykład

Dla danych wejściowych:

```
7 7
1 2
1 3
2 3
3 4
4 5
4 5
7 6
```

poprawnym wynikiem jest:

```
4
><>>><<
```



**Wyjaśnienie do przykładu:** Na rysunku przedstawiono pewne optymalne skierowanie ulic, dla którego mamy cztery osiedla utworzone przez następujące zbiory skrzyżowań:  $\{1, 2, 3\}$ ,  $\{4, 5\}$ ,  $\{6\}$  i  $\{7\}$ . Zauważ, że np. choć można dojechać ze skrzyżowania 3 do skrzyżowania 4, to nie należą one do tego samego osiedla, bo nie można dojechać ze skrzyżowania 4 do skrzyżowania 3.

### Testy „ocen”:

**1ocen:**  $n = 7$ ,  $m = 10$  – mały test poprawnościowy;

**2ocen:**  $n = 5000$ ,  $m = 4999$  – ścieżka (dla każdego skrzyżowania  $i \in \{1, \dots, n-1\}$  jest ulica między  $i$  a  $i+1$ );

**3ocen:**  $n = 2000$ ,  $m = 20\,000$  – dziesięciokrotny cykl (dla każdego skrzyżowania  $i$  jest dokładnie dziesięć ulic między  $i$  a  $(i+1) \bmod n$ );

**4ocen:**  $n = 500\,000$ ,  $m = 999\,998$  – dla każdego skrzyżowania  $i \in \{1, \dots, n-2\}$  są ulice między  $i$  a  $i+1$  oraz między  $i$  a  $i+2$ ; dodatkowo są dwie ulice między  $n-1$  a  $n$ .

## Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Limity czasowe obowiązujące w poszczególnych podzadaniach są opublikowane w SIO.

Podzadanie	Warunki	Liczba punktów
1	$n, m \leq 5000$	16
2	$n \leq 2000, m \leq 20\,000$	12
3	$n \leq 5000$	20
4	brak dodatkowych ograniczeń	52

Jeżeli Twój program wypisze jedynie jeden z wierszy na wyjściu poprawnie, wciąż otrzyma on 50% za dany test. Jeśli to drugi wiersz jest poprawny, musisz zadbać o to, aby Twój program wypisał dwa wiersze, przy czym pierwszy musi zawierać 32-bitową liczbę całkowitą (typu `int`).