

Zadanie: ROW

Równoważne programy



XXIII OI, etap III, dzień 0. Plik źródłowy row.* Dostępna pamięć: 128 MB.

12.04.2016

Bajtazar dostał nowy komputer i uczy się go programować. Program składa się z ciągu instrukcji. Jest k różnych rodzajów instrukcji, które dla uproszczenia oznaczamy liczbami od 1 do k . Niektóre pary instrukcji mają tę własność, że jeśli występują w programie bezpośrednio obok siebie (w dowolnej kolejności), to zamieniając je miejscami, nie zmienia się działania programu (czyli uzyskuje się program *równoważny*). Pozostałe pary instrukcji nie mają tej własności i nazywamy je parami *nieprzemiennymi*. Bajtazar napisał dwa programy o długości n instrukcji każdy i zastanawia się, czy są one równoważne. Pomóż mu!

Wejście

W pierwszym wierszu standardowego wejścia znajdują się trzy liczby całkowite n , k oraz m podzielane pojedynczymi odstępami, oznaczające odpowiednio długość programów, liczbę różnych instrukcji komputera oraz liczbę par instrukcji nieprzemiennych.

Kolejne m wierszy zawiera opis tych par: każdy z tych wierszy zawiera dwie liczby całkowite a i b ($1 \leq a < b \leq k$) oddzielone pojedynczym odstępem, oznaczające, że para instrukcji o numerach a i b jest nieprzemienna. Możesz założyć, że każda para wystąpi w tym opisie co najwyżej raz.

Kolejne dwa wiersze przedstawiają opisy dwóch programów. Każdy z tych wierszy zawiera ciąg n liczb całkowitych c_1, c_2, \dots, c_n ($1 \leq c_i \leq k$) podzielanych pojedynczymi odstępami, oznaczających numery kolejnych instrukcji programu.

Wyjście

W jedynym wierszu standardowego wyjścia należy wypisać jedno słowo TAK lub NIE w zależności od tego, czy podane na wejściu programy są równoważne.

Przykład

Dla danych wejściowych:

```
5 3 1
2 3
1 1 2 1 3
1 2 3 1 1
```

poprawnym wynikiem jest:

TAK

natomiast dla danych wejściowych:

```
3 3 1
2 3
1 2 3
3 2 1
```

poprawnym wynikiem jest:

NIE

Wyjaśnienie do pierwszego przykładu: W pierwszym programie można zamienić instrukcje na pozycjach 2 i 3, a następnie instrukcję na pozycji 5 z instrukcjami na pozycjach 4 i 3. W ten sposób uzyska się drugi program.

Testy „ocen”:

1ocen: $n = 50$, $k = 50$, $m = 1$; programy to $(1, 2, \dots, 49, 50)$ oraz $(50, 49, \dots, 2, 1)$; odpowiedź NIE.

2ocen: $n = 99\,999$, $k = 3$, $m = 1$; instrukcje nieprzemienne to 1 i 2, a programy to $(1, 2, 3, 1, 2, 3, \dots, 1, 2, 3)$ oraz $(3, 1, 2, 3, 1, 2, \dots, 3, 1, 2)$; odpowiedź TAK.

3ocen: $n = 100\,000$, $k = 1000$, $m = 50\,000$; programy to $(13, 13, 13, \dots, 13)$ oraz $(37, 37, 37, \dots, 37)$; odpowiedź to oczywiście NIE.

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów. We wszystkich testach zachodzą warunki $1 \leq n \leq 100\,000$, $1 \leq k \leq 1000$, $0 \leq m \leq 50\,000$.

Podzadanie	Warunki	Liczba punktów
1	$n \leq 5$	5
2	$k \leq 2$	5
3	$n \leq 1000$	25
4	brak dodatkowych warunków	65