

Boty

XVII OIJ, zawody III stopnia – dzień drugi
7 maja 2023

Kod zadania: **bot**
Limit czasu: **4 s (C++) / 60 s (Python)**
Limit pamięci: **256 MB**



Bajtek gra w grę strategiczną. Lubi wiele z jej aspektów, ale niestety, nudzi go zbieranie zasobów. Dlatego przygotował *bota* (mini-programik, który symuluje ruchy gracza w grze), który chodzi po mapie, odwiedza różne lokalizacje i zbiera zasoby.

Bajtek nie potrafi jeszcze zbyt dobrze programować, dlatego jego bot jest bardzo prosty. Dla każdej lokalizacji Bajtek ustalił w kodzie źródłowym bota lokalizację, do której bot przejdzie w kolejnym ruchu. Może się zdarzyć, że bot przejdzie do tej samej lokalizacji. Uruchamiając bota, Bajtek podaje mu lokalizację, w której bot zaczyna swoją podróż. Jedno przejście pomiędzy dowolnymi lokalizacjami zajmuje botowi jedną sekundę, a czas zbierania zasobów jest pomijalny.

Początkowo Bajtek był zadowolony ze swojego dzieła, jednak po pewnym czasie doszedł do wniosku, że bot zbiera zasoby za wolno, a być może wręcz niektóre pomija. Wtedy wpadł na genialny pomysł! Przecież po coś ma wielordzeniowy procesor! Może więc uruchomić wiele botów na raz, a dokładniej jest w stanie uruchomić jednocześnie M botów. Może system wykrywania oszustw nie wykryje jego małego oszustwa, jeśli uruchomi niewiele botów jednocześnie?

Bajtek chce, żeby jego boty odwiedziły wszystkie lokalizacje na mapie co najmniej raz, żeby zebrać wszystkie zasoby, i żeby zajęło to jak najmniej czasu. Czy pomożesz mu optymalnie rozmieścić jego boty, żeby tak się stało?

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna N ($1 \leq N \leq 200\,000$) określająca liczbę lokalizacji na mapie. W drugim wierszu znajduje się ciąg N liczb naturalnych A_i ($1 \leq A_i \leq N$) pooddzielanych pojedynczymi odstępami, i -ta liczba określa numer lokalizacji, do której bot przejdzie z lokalizacji nr i . W trzecim (ostatnim) wierszu wejścia znajduje się jedna liczba naturalna M ($1 \leq M \leq 200\,000$) – liczba botów, które może uruchomić Bajtek.

Lokalizacje numerowane są kolejnymi liczbami naturalnymi od 1 do N .

Wyjście

Jeżeli nie jest możliwe takie rozmieszczenie co najwyżej M botów, żeby zebrały one wszystkie zasoby, na wyjście należy wypisać jedno słowo NIE.

W przeciwnym wypadku, w pierwszym wierszu wyjścia należy wypisać jedną liczbę całkowitą – minimalny czas, w jakim optymalnie rozmieszczone boty zbiorą wszystkie zasoby. W drugim wierszu wyjścia należy wypisać jedną liczbę całkowitą B – liczbę rozmieszczonych botów. W trzecim (ostatnim) wierszu wyjścia należy wypisać ciąg B liczb naturalnych określających numery lokalizacji, z których powinny startować kolejne boty.

Jeżeli istnieje wiele możliwych rozwiązań, Twój program może wypisać dowolne z nich.

Ocenianie

Możesz rozwiązać zadanie w kilku prostszych wariantach – niektóre grupy testów spełniają pewne dodatkowe ograniczenia. Poniższa tabela pokazuje, ile punktów otrzyma Twój program, jeśli przejdzie testy z takim ograniczeniem.

Dodatkowe ograniczenia	Liczba punktów
$M = 200\,000$	8
$N \leq 18$	25
$N \leq 1\,000$	60



Przykłady

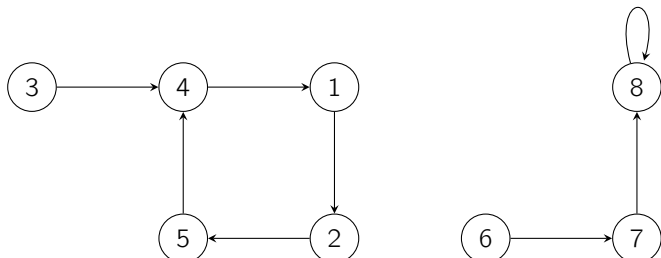
Wejście dla testu bot0a:

```
8
2 5 4 1 4 7 8 8
3
```

Wyjście dla testu bot0a:

```
2
3
3 2 6
```

Mapę gry z tego przykładu można zilustrować na schemacie:



Bajtek może umieścić swoje boty w lokalizacjach nr 3, 2 i 6. Bot umieszczony w lokalizacji nr 3 po sekundzie przejdzie do lokalizacji nr 4 a po kolejnej sekundzie do lokalizacji nr 1. Bot umieszczony w lokalizacji nr 2 w tym samym czasie przejdzie do lokalizacji nr 5 i 4, zaś bot umieszczony w lokalizacji nr 6 przejdzie do lokalizacji nr 7 i 8. Zatem, po dwóch sekundach boty odwiedzą wszystkie lokacje i zbiorą z nich wszystkie zasoby.

Wejście dla testu bot0b:

```
4
2 3 2 3
1
```

Wyjście dla testu bot0b:

```
NIE
```

W tym przykładzie Bajtek nie ma możliwości umieszczenia jednego bota, tak żeby zebrał on zasoby ze wszystkich lokalizacji.

Wejście dla testu bot0c:

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

Wyjście dla testu bot0c:

```
3
2
1 5
```

W tym przykładzie Bajtek może umieścić boty w lokalizacjach nr 1 i nr 5. Wtedy, w czasie trzech sekund, pierwszy bot zbierze zasoby z lokalizacji o numerach od 1 do 4, a drugi bot z lokalizacji o numerach od 5 do 7.

Pozostałe testy przykładowe

- test bot0d: $N = 100$, $M = 10$, z lokalizacji nr i (dla $i < N$) bot przechodzi do lokalizacji nr $i + 1$, a z lokalizacji nr N przechodzi do lokalizacji nr 1;
- test bot0e: $N = M = 200\,000$, z każdej lokalizacji bot przechodzi do niej samej.