

Dynamiczna permutacja (dynamiczna-permutacja)

Limit pamięci: 64 MB

Limit czasu: 5.00 s

W tym zadaniu celem jest przygotowanie algorytmu/struktury danych do zarządzania dynamiczną permutacją: pewnym ustawieniem liczb $1, 2, \dots, N$ w ciąg.

Na początku otrzymujemy ciąg π : ciąg parami różnych liczb $\pi[1], \pi[2], \dots, \pi[N]$ z przedziału 1 do N . Następnie, należy obsłużyć operacje/zapytania następujących typów:

- zamień $\pi[i]$ oraz $\pi[j]$,
- wyznacz największą wartość spośród $z, \pi[z], \pi[\pi[z]], \pi[\pi[\pi[z]]], \dots, \pi^k[z]$.

Napisz program, który wczyta początkową permutację oraz operacje i zapytania, wyznaczy odpowiedzi na wszystkie zapytania i wypisze wyniki na standardowe wyjście.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna N , oznaczająca długość permutacji. W drugim wierszu wejścia znajduje się N parami różnych liczb naturalnych π_i , pooddzielanych pojedynczymi odstępami. W trzecim wierszu wejścia znajduje się jedna liczba naturalna Q , oznaczająca liczbę operacji/zapytań. W kolejnych Q wierszach znajdują się operacje/zapytania, po jednym w wierszu. Każde zapytanie jest postaci:

- swap $x_i y_i$ – dla wartości $1 \leq x_i, y_i \leq N$, wykonaj zamianę $\pi[x_i]$ oraz $\pi[y_i]$,
- query $z_i k_i$ – dla wartości $1 \leq z_i \leq N$, podaj największą wartość ze zbioru $\{z_i, \pi[z_i], \pi[\pi[z_i]], \dots, \pi^{k_i}[z_i]\}$.

Wyjście

Dla każdego zapytania query, zgodnie z kolejnością na wejściu należy wypisać odpowiedź. Odpowiedzi dla zapytań należy wypisywać w osobnych wierszach, bez dodatkowych odstępów.

Ograniczenia

$1 \leq N \leq 100\,000, 1 \leq Q \leq 100\,000, 1 \leq k_i \leq N$.

Przykład

Wejście

```
5
2 3 1 5 4
7
query 2 3
query 2 1
swap 3 5
query 2 3
query 1 5
swap 5 1
query 2 1
```

Wyjście

```
3
3
5
5
3
```

Kwadrat magiczny (kwadrat-magiczny)

Limit pamięci: 256 MB

Limit czasu: 4.00 s

Jasio w czasopiśmie z zagadkami odnalazł zagadkę uzupełnienia pól kwadratu magicznego. Bardzo spodobała mu się ta zagadka oraz ogólnie koncept kwadratów magicznych. Kwadrat magiczny to plansza składająca się z $N \times N$ pól. W każdym polu znajduje się liczba naturalna. Wszystkie sumy liczb w każdym rzędzie i w każdej kolumnie są sobie równe.

Jasio chciałby stworzyć swój własny kwadrat magiczny. Okazuje się nie być to takie proste. Zawsze jakaś suma mu się nie zgadza, a zmieniając pojedynczą liczbę Jasio psuje inne sumy, które już się ze sobą zgadzały. Celem Jasia jest, żeby suma każdego wiersza i każdej kolumny była równa dokładnie M . Postanowił poprosić Cię o pomoc w zamienieniu liczb z niektórych pól jego planszy na zera, aby to osiągnąć. Czy pomożesz mu w tym zadaniu?

Napisz program, który: wczyta opis planszy Jasia oraz założoną przez niego wartość M , wyznaczy zbiór pól, które należy zamienić na zera, aby osiągnąć kwadrat magiczny, w którym suma każdego wiersza i suma każdej kolumny jest równa dokładnie M i wypisze wynik na standardowe wyjście.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby naturalne N oraz M , oddzielone pojedynczym odstępem. Określają one kolejno rozmiar planszy oraz oczekiwaną przez Jasia sumę każdego wiersza i kolumny. W kolejnych N wierszach znajduje się po N liczb naturalnych $A_{i,j}$ pooddzielanych pojedynczymi odstępami. Są to liczby zapisane na planszy Jasia.

Możesz założyć, że testy są tak dobrane, że każdy ma co najmniej jedno rozwiązanie. Jeżeli istnieje wiele poprawnych rozwiązań, Twój program może wypisać dowolne z nich.

Wyjście

Twój program powinien wypisać na wyjście N wierszy, w każdym z nich po N liczb pooddzielanych pojedynczymi odstępami – planszę reprezentującą kwadrat magiczny po zmianie niektórych wartości na zera.

Ograniczenia

$1 \leq N \leq 12$, $1 \leq M \leq 10\,000$, $1 \leq A_{i,j} \leq 10\,000$.

Przykład

Wejście

```
3 10
6 5 4
2 8 7
2 2 6
```

Wyjście

```
6 0 4
2 8 0
2 2 6
```

Mediana podzbioru (mediana-podzbioru)

Limit pamięci: 256 MB

Limit czasu: 8.00 s

Na taśmie o wymiarach $N \times 1$ pól znajdują się liczby naturalne, po jednej liczbie na jednym polu taśmy.

Twoim zadaniem jest (efektywnie) obsłużyć wiele zapytań o różne spójne fragmenty tej taśmy. W każdym zapytaniu rozpatrywany jest fragment od S_i -tego do E_i -tego pola taśmy włącznie. Pytanie dotyczy tego jaka jest mediana **zbioru** $\{T[S_i], T[S_i + 1], \dots, T[E_i]\}$. Zwróć uwagę na pogrubione słowo **zbioru**: powtórzone elementy na taśmie występują w zbiorze jedynie jeden raz.

Dla przypomnienia: mediana zbioru M -elementowego to średnia arytmetyczna co najwyżej dwóch środkowych elementów zbioru po jego posortowaniu: jeżeli elementy zbioru to $X[1] < X[2] < \dots < X[M]$ to mediana wynosi $\frac{X[\lfloor (M+1)/2 \rfloor] + X[\lceil (M+1)/2 \rceil]}{2}$.

Napisz program, który: wczyta liczby zapisane na taśmie oraz zapytania, dla każdego zapytania ustali medianę odpowiedniego podzbioru liczb z taśmy i wypisze wyniki na standardowe wyjście.

Wejście

W pierwszym wierszu wejścia znajduje się jedna liczba naturalna N , określająca długość taśmy. W drugim wierszu wejścia znajduje się ciąg N liczb naturalnych $T[i]$, pooddzielanych pojedynczymi odstępami. Są to liczby zapisane na taśmie. W trzecim wierszu wejścia znajduje się jedna liczba naturalna Q , określająca liczbę zapytań. W kolejnych Q wierszach znajduje się opis kolejnych zapytań, po jednym w wierszu.

Opis każdego zapytania składa się z dwóch liczb naturalnych S_i oraz E_i oddzielonych pojedynczym odstępem. Jest to zapytanie o medianę zbioru $\{T[S_i], T[S_i + 1], \dots, T[E_i]\}$.

Wyjście

Twój program powinien wypisać na wyjście dokładnie Q wierszy. W i -tym wierszu wyjścia powinna się znaleźć odpowiedź dla i -tego zapytania: liczba rzeczywista, wypisana z dokładnością do jednego miejsca po kropce dziesiętnej, reprezentująca medianę podzbioru z danego zapytania.

Ograniczenia

$1 \leq N \leq 200\,000$, $1 \leq Q \leq 200\,000$, $1 \leq T[i] \leq 10^9$, $1 \leq S_i \leq E_i \leq N$.

Częściowa punktacja

W pewnym podzbiore testów wartych łącznie 50% maksymalnej punktacji zachodzi dodatkowy warunek: wszystkie liczby na taśmie są parami różne.

Przykład

Wejście

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

Wyjście

```
3.0
3.5
4.0
```