

Равни мех-ове

Добре известно е сред румънските знатни люде, че красотата на масив от цели числа $a[0], a[1], a[2], \dots, a[m-1]$ се състои в точния брой положителни цели числа k , за които масивът може да бъде разбит на k неприсичащи се подмасиви (редици от последователно разположени елементи), така че всеки елемент се съдържа в точно един подмасив и всички подмасиви имат равен мех (minimum excluded element). Мех на целочислен масив е най-малкото положително число (**обърнете внимание - по-голямо от 0**), което не се среща в масива.

Дадени са Ви целочислен масив $v[0], v[1], \dots, v[n-1]$ и q заявки във вида (l_i, r_i) , където $0 \leq l_i \leq r_i < n$ за всяко $0 \leq i < q$.

За всяка заявка трябва да намерите красотата на масива $v[l_i], v[l_i + 1], \dots, v[r_i]$.

Детайли по имплементацията

Трябва да имплементирате следната функция:

```
std::vector<int> solve(  
    int n, std::vector<int>& v,  
    int q, std::vector<std::pair<int, int>>& queries);
```

- n : размерът на целочисления масив
- v : вектор с размер n , началният масив
- q : броят на заявките
- $queries$: вектор с размер q , описващ заявките
- Тази функция трябва да върне вектор от q цели числа, съдържащ отговорите на заявките.
- Тази функция се извиква точно веднъж за всеки тест.

Ограничения

- $1 \leq n \leq 600\,000$
- $1 \leq q \leq 600\,000$
- $1 \leq v[i] \leq 400\,000$ за всяко $0 \leq i < n$
- $0 \leq l_i \leq r_i < n$ за всяко $0 \leq i < q$

Подзадачи

1. (4 точки) $1 \leq n \leq 10, 1 \leq q \leq 100$
2. (6 точки) $1 \leq n, q \leq 100$
3. (17 точки) $1 \leq n, q \leq 1\,000$
4. (10 точки) $1 \leq n, q \leq 100\,000$ и $1 \leq v[i] \leq 2$ за всяко $0 \leq i < n$
5. (30 точки) $1 \leq n, q \leq 75\,000$
6. (33 точки) Няма допълнителни ограничения.

Пример

Нека разгледаме следното извикване:

```
solve(10, {1, 1, 2, 2, 3, 3, 1, 2, 3, 4}, 2, {{0, 5}, {0, 8}})
```

В този пример $n = 10$ и има 2 заявки, за които:

- $l_0 = 0$ и $r_0 = 5$
- $l_1 = 0$ и $r_1 = 8$

За първата заявка можем да разбием масива само по един начин - като подмасива от индекс 0 до индекс 5.

За втората заявка, k може да е или 1, или 2.

При разбиването на 1 подмасив това е просто подмасивът от индекс 0 до индекс 8.

При разбиването на 2 подмасива една възможност е на подмасива от индекс 0 до индекс 5 и подмасива от индекс 6 до индекс 8.

Отговорът на първата заявка е 1, а на втората заявка е 2, така че `solve` трябва да върне вектора $\{1, 2\}$.

Локален грейдър

Локалният грейдър чете входа в следния формат:

- ред 1: $n\ q$
- ред 2: $v[0]\ v[1]\ \dots\ v[n-1]$
- ред $3 + i$: $l_i\ r_i$ за всяко $0 \leq i < q$

и извежда q реда с намерените отговори от извикването на функцията `solve` със съответните параметри.