

Equalmex

Familiile nobile din România cunosc foarte bine faptul că frumusețea unui array de întregi $a[0], a[1], a[2], \dots, a[m-1]$ este cantitatea de numere întregi pozitive k pentru care puteți diviza array-ul în k subarray-uri (secvențe de elemente consecutive) disjuncte, astfel încât fiecare element se conține în exact un subarray și toate subarray-urile au același element exclus minimal. Elementul exclus minimal al unui array de întregi este cel mai mic număr strict pozitiv (**mai mare decât** 0) care nu apare în array.

Vă este dat un array de întregi $v[0], v[1], \dots, v[n-1]$ și q interogări de forma (l_i, r_i) , unde $0 \leq l_i \leq r_i < n, 0 \leq i < q$.

Pentru fiecare interogare, urmează să determinați frumusețea array-ului $v[l_i], v[l_i + 1], \dots, v[r_i]$.

Detalii de Implementare

Urmează să implementați următoarea funcție:

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

- n : dimensiunea array-ului de numere întregi
- v : array de lungimea n , array-ul inițial
- q : numărul de interogări
- $queries$: array de lungime q cu descrieri ale interogărilor
- Această funcție trebuie să returneze un array de q numere întregi conținând răspunsul pentru fiecare interogare
- Această funcție va fi apelată exact o singură dată pentru fiecare test

Restricții

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

Subtask-uri

1. (4 puncte) $1 \leq n \leq 10, 1 \leq q \leq 100$
2. (6 puncte) $1 \leq n, q \leq 100$
3. (17 puncte) $1 \leq n, q \leq 1\,000$
4. (10 puncte) $1 \leq n, q \leq 100\,000$ și $1 \leq v[i] \leq 2$ pentru toți $0 \leq i < n$
5. (30 puncte) $1 \leq n, q \leq 75\,000$
6. (33 puncte) Fără restricții adiționale.

Exemple

Exemplul 1

Considerați următorul apel:

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

În acest exemplu $n = 10$ și există 2 interogări pentru care:

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

Pentru prima interogare, putem împărți intervalul în doar o singură subsecvență, care acoperă pozițiile de la 0 până la 5.

În cea de a doua interogare, k poate fi atât 1 cât și 2.

O posibilitate de divizare în 1 subarray este selectarea subarray-ului de la poziția 0 la poziția 8. O posibilitate de divizare în 2 subarray-uri este prin selectarea unui subarray de la poziția 0 la poziția 5 și de la poziția 6 la poziția 8.

Răspunsul pentru prima interogare este 1, și pentru cea de a doua 2, prin urmare funcția `solve` va returna `{1, 2}`.

Grader Local

Graderul Local citește date de input în următorul format:

- linia 1: $n\ q$
- linia 2: $v[0]\ v[1]\ \dots\ v[n-1]$
- linia $3 + i$: $l_i\ r_i$ pentru toți $0 \leq i < q$

și afișează q linii, rezultatul apelului funcției `solve` cu parametrii corespunzătoare.