Equalmex

Серед румунської знаті добре відомо, що краса цілочисельного масиву $a[0], a[1], a[2], \ldots a[m-1]$ полягає в кількості додатних цілих чисел k, для яких можна розбити масив на k неперетинних підмасивів (послідовностей з послідовних елементів) так, щоб кожен елемент входив рівно в один підмасив, і всі підмасиви мали однакове мінімальне виключене значення. Мінімальне виключене значення (mex) цілочислового масиву — це найменше строго додатне ціле число (більше за 0), яке не зустрічається в масиві.

Вам задано цілочисельний масив $v[0], v[1], \ldots v[n-1]$ та q запитів виду (l_i, r_i), де $0 \leq l_i \leq r_i < n$ для всіх $0 \leq i < q$.

Для кожного запиту вам потрібно знайти «красу» масиву $v[l_i], v[l_i+1], \ldots, v[r_i]$.

Деталі реалізації

Потрібно реалізувати наступну функцію:

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

- n: розмір цілочисельного масиву
- v: масив довжини n, початковий масив
- q: кількість запитів
- $queries$: масив довжини q, що описує запити
- Ця функція повинна повертати вектор із q цілих чисел, що містить відповідь на кожен запит.
- Ця функція викликається рівно один раз для кожного тестового випадку.

Обмеження

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

Підзадачі

1. (4 бали) $1 \le n \le 10, 1 \le q \le 100$
2. (6 балів) $1 \le n, q \le 100$
3. (17 балів) $1 \le n, q \le 1\,000$
4. (10 балів) $1 \le n, q \le 100\,000$ та $1 \le v[i] \le 2$ для всіх $0 \le i < n$
5. (30 балів) $1 \le n, q \le 75\,000$
6. (33 бали) Без додаткових обмежень.

Приклади

Приклад 1

Розглянемо наступний виклик:

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

У цьому прикладі $n = 10$, і є 2 запити, для яких:

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

Для першого запиту ми можемо розділити інтервал лише на один підмасив, який знаходиться від позиції `0` до позиції `5`.

У другому запиті k може бути або 1, або 2. Можливість розбиття на 1 підмасив полягає у виборі підмасиву від позиції `0` до позиції `8`. Можливість розбиття на 2 підмасиви є вибір підмасиву з позиції `0` до позиції `5` та підмасиву з позиції `6` до позиції `8`.

Відповіддю на перший запит буде 1, а на другий запит буде 2, тому виклик `solve` поверне `{1, 2}`.

Приклад градера

Зчитує вхідні дані у такому форматі:

- рядок 1: $n\ q$
- рядок 2: $v[0]\ v[1]\ \ldots\ v[n-1]$
- рядок $3+i$: $l_i\ r_i$ для всіх $0 \le i < q$

і виводить q рядків, результат виклику функції `solve` з відповідними параметрами.