

## Równomex

Jak powszechnie wiadomo wśród rumuńskiej arystokracji, piękno tablicy liczb całkowitych  $a[0], a[1], a[2], \dots, a[m-1]$  to liczba dodatnich liczb całkowitych  $k$ , takich że można tę tablicę podzielić na  $k$  rozłącznych podtablic (ciągów kolejnych elementów), takich, że każdy element należy do dokładnie jednej podtablicy oraz wszystkie podtablice mają ten sam mex. Mex tablicy liczb całkowitych to najmniejsza dodatnia liczba całkowita (większa niż 0), która nie występuje w tej tablicy.

Dana jest tablica liczb całkowitych  $v[0], v[1], \dots, v[n-1]$  i  $q$  zapytań postaci  $(l_i, r_i)$ , gdzie  $0 \leq l_i \leq r_i < n$  dla  $0 \leq i < q$ .

Dla każdego zapytania, znajdź piękno tablicy  $v[l_i], v[l_i + 1], \dots, v[r_i]$ .

## Szczegóły implementacji

Zaimplementuj następującą funkcję:

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

- $n$ : rozmiar tablicy liczb całkowitych
- $v$ : wektor rozmiaru  $n$  z daną tablicą.
- $q$ : liczba zapytań
- $queries$ : wektor rozmiaru  $q$  z zapytaniami
- Funkcja ma zwrócić wektor złożony z  $q$  liczb całkowitych będących odpowiedziami na zapytania.
- Ta funkcja zostanie wywołana jeden raz dla każdego testu.

## Ograniczenia

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

## Podzadania

1. (4 punkty)  $1 \leq n \leq 10, 1 \leq q \leq 100$
2. (6 punktów)  $1 \leq n, q \leq 100$
3. (17 punktów)  $1 \leq n, q \leq 1\,000$
4. (10 punktów)  $1 \leq n, q \leq 100\,000$  oraz  $1 \leq v[i] \leq 2$  dla każdego  $0 \leq i < n$
5. (30 punktów)  $1 \leq n, q \leq 75\,000$
6. (33 punktów) Bez dodatkowych ograniczeń.

## Przykłady

### Przykład 1

Rozważmy następujące wywołanie:

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

W tym przykładzie  $n = 10$  i zadane są 2 zapytania, w których:

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

Dla pierwszego zapytania, możemy podzielić wskazany wycinek tablicy tylko na jedną podtablicę od pozycji 0 do pozycji 5.

Przy drugim zapytaniu  $k$  może być 1 lub 2. Podział na 1 podtablicę oznacza wybranie podtablicy od pozycji 0 do pozycji 8. Podział na 2 podtablice oznacza wybranie podtablicy od pozycji 0 do pozycji 5 i podtablicy od pozycji 6 do pozycji 8.

Odpowiedzią na pierwsze zapytanie jest 1 a na drugie 2, więc wywołanie `solve` zwróci  $\{1, 2\}$ .

## Przykładowa sprawdzaczka

Przykładowa sprawdzaczka odczytuje dane w następującym formacie:

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

i wypisuje  $q$  linii, tj. wynik wywołania funkcji `solve` z zadanymi parametrami.