2936. Number of Equal Numbers Blocks Premium

Medium | Topics | Companies | Hint

You are given a **0-indexed** array of integers, `nums`. The following property holds for `nums`:

- All occurrences of a value are adjacent. In other words, if there are two indices `i < j` such that `nums[i] == nums[j]`, then for every index `k` that `i < k < j`, `nums[k] == nums[i]`.

Since `nums` is a very large array, you are given an instance of the class `BigArray` which has the following functions:

- `int at(long long index)`: Returns the value of `nums[i]`.
- `void size()`: Returns `nums.length`.

Let's partition the array into **maximal** blocks such that each block contains **equal values**. Return *the number of these blocks.*

Note that if you want to test your solution using a custom test, behavior for tests with `nums.length > 10` is undefined.

Example 1:

```
Input: nums = [3,3,3,3,3]
Output: 1
Explanation: There is only one block here which is the whole array (because all numbers are equal) and that is: [3,3,3,3,3]. So the answer would be 1.
```

Example 2:

```
Input: nums = [1,1,1,3,9,9,9,2,10,10]
Output: 5
Explanation: There are 5 blocks here:
Block number 1: [1,1,1,3,9,9,9,2,10,10]
Block number 2: [1,1,1,3,9,9,9,2,10,10]
Block number 3: [1,1,1,3,9,9,9,2,10,10]
Block number 4: [1,1,1,3,9,9,9,2,10,10]
Block number 5: [1,1,1,3,9,9,9,2,10,10]
So the answer would be 5.
```

Example 3:

```
Input: nums = [1,2,3,4,5,6,7]
Output: 7
Explanation: Since all numbers are distinct, there are 7 blocks here and each element representing one block. So the answer would be 7.
```

Constraints:

- `1 <= nums.length <= 10^15`
- `1 <= nums[i] <= 10^9`
- The input is generated such that all equal values are adjacent.
- The sum of the elements of `nums` is at most `10^15`.

Seen this question in a real interview before? 1/5

Yes No

Accepted 1.9K | Submissions 3K | Acceptance Rate 64.1%

Topics

Array Binary Search Interactive

Companies

0 - 6 months

Google 2

Hint 1

Start from the beginning of the array `nums.at(0)`.

Hint 2

Do a binary search on the last index `last` such that `nums.at(0) == nums.at(last)`.

Hint 3

Continue this process until you get to the end of the array (`nums.size()`).

Discussion (5)