Problem 2936: Number of Equal Numbers Blocks

Problem Information

Difficulty: Medium
Acceptance Rate: 62.34%
Paid Only: Yes
Tags: Array, Binary Search, Interactive

Problem Description

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 <= 1015` * `1 <= nums[i] <= 109` * The input is generated such that all equal values are adjacent. * The sum of the elements of `nums` is at most `1015`.

Code Snippets

C++:

```
/**
* Definition for BigArray.
* class BigArray {
* public:
* BigArray(vector<int> elements);
* int at(long long index);
* long long size();
* };
*/
class Solution {
public:
int countBlocks(BigArray* nums) {

}
};
```

Java:

```
/**
* Definition for BigArray.
* class BigArray {
```

```
* public BigArray(int[] elements);
* public int at(long index);
* public long size();
* }
*/
class Solution {
public int countBlocks(BigArray nums) {

}
}
```

Python3:

```
# Definition for BigArray.
# class BigArray:
# def at(self, index: long) -> int:
# pass
# def size(self) -> long:
# pass
class Solution(object):
def countBlocks(self, nums: Optional['BigArray']) -> int:
```