Problem 3113: Find the Number of Subarrays Where Boundary Elements Are Maximum

Problem Information

Difficulty: Hard
Acceptance Rate: 32.10%
Paid Only: No
Tags: Array, Binary Search, Stack, Monotonic Stack

Problem Description

You are given an array of **positive** integers `nums`.

Return the number of subarrays of `nums`, where the **first** and the **last** elements of the subarray are _equal_ to the **largest** element in the subarray.

Example 1:

Input: nums = [1,4,3,3,2]

Output: 6

Explanation:

There are 6 subarrays which have the first and the last elements equal to the largest element of the subarray:

* subarray `[**_1_** ,4,3,3,2]`, with its largest element 1. The first element is 1 and the last element is also 1. * subarray `[1,_**4**_ ,3,3,2]`, with its largest element 4. The first element is 4 and the last element is also 4. * subarray `[1,4,_**3**_ ,3,2]`, with its largest element 3. The first element is 3 and the last element is also 3. * subarray `[1,4,3,_**3**_ ,2]`, with its largest element 3. The first element is 3 and the last element is also 3. * subarray `[1,4,3,3,_**2**_]`, with its largest element 2. The first element is 2 and the last element is also 2. * subarray `[1,4,_**3,3**_ ,2]`, with its largest element 3. The first element is 3 and the last element is also 3.

Hence, we return 6.

Example 2:

Input: nums = [3,3,3]

Output: 6

Explanation:

There are 6 subarrays which have the first and the last elements equal to the largest element of the subarray:

* subarray `[_**3**_ ,3,3]`, with its largest element 3. The first element is 3 and the last element is also 3. * subarray `[3,**_3_** ,3]`, with its largest element 3. The first element is 3 and the last element is also 3. * subarray `[3,3,_**3**_]`, with its largest element 3. The first element is 3 and the last element is also 3. * subarray `[**_3,3_** ,3]`, with its largest element 3. The first element is 3 and the last element is also 3. * subarray `[3,_**3,3**_]`, with its largest element 3. The first element is 3 and the last element is also 3. * subarray `[_**3,3,3**_]`, with its largest element 3. The first element is 3 and the last element is also 3.

Hence, we return 6.

Example 3:

Input: nums = [1]

Output: 1

Explanation:

There is a single subarray of `nums` which is `[**_1_**]`, with its largest element 1. The first element is 1 and the last element is also 1.

Hence, we return 1.

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 109`

Code Snippets

C++:

```
class Solution {
public:
long long numberOfSubarrays(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public long numberOfSubarrays(int[] nums) {

}
}
```

Python3:

```
class Solution:
def numberOfSubarrays(self, nums: List[int]) -> int:
```