Problem 3739: Count Subarrays With Majority Element II

Problem Information

Difficulty: Hard
Acceptance Rate: 41.78%
Paid Only: No
Tags: Array, Hash Table, Divide and Conquer, Segment Tree, Merge Sort, Prefix Sum

Problem Description

You are given an integer array `nums` and an integer `target`.

Return the number of **subarrays** of `nums` in which `target` is the **majority element**.

The **majority element** of a subarray is the element that appears **strictly more than half** of the times in that subarray.

Example 1:

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

Output: 5

Explanation:

Valid subarrays with `target = 2` as the majority element:

* `nums[1..1] = [2]` * `nums[2..2] = [2]` * `nums[1..2] = [2,2]` * `nums[0..2] = [1,2,2]` * `nums[1..3] = [2,2,3]`

So there are 5 such subarrays.

Example 2:

Input: nums = [1,1,1,1], target = 1

Output: 10

Explanation:

■■■■■■■■ All 10 subarrays have 1 as the majority element.

Example 3:

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

Output: 0

Explanation:

`target = 4` does not appear in `nums` at all. Therefore, there cannot be any subarray where 4 is the majority element. Hence the answer is 0.

Constraints:

* `1 <= nums.length <= 10■■■■■■■5` * `1 <= nums[i] <= 10■■■■■■■9` * `1 <= target <= 109`

Code Snippets

C++:

```
class Solution {
public:
long long countMajoritySubarrays(vector<int>& nums, int target) {

}
};
```

Java:

```
class Solution {
public long countMajoritySubarrays(int[] nums, int target) {
```

```
}
}
```

Python3:

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