Problem 3101: Count Alternating Subarrays

Problem Information

Difficulty: Medium
Acceptance Rate: 56.98%
Paid Only: No
Tags: Array, Math

Problem Description

You are given a binary array `nums`.

We call a subarray **alternating** if **no** two **adjacent** elements in the subarray have the **same** value.

Return _the number of alternating subarrays in_`nums`.

Example 1:

Input: nums = [0,1,1,1]

Output: 5

Explanation:

The following subarrays are alternating: `[0]`, `[1]`, `[1]`, `[1]`, and `[0,1]`.

Example 2:

Input: nums = [1,0,1,0]

Output: 10

Explanation:

Every subarray of the array is alternating. There are 10 possible subarrays that we can choose.

Constraints:

* `1 <= nums.length <= 105` * `nums[i]` is either `0` or `1`.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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