Problem 3392: Count Subarrays of Length Three With a Condition

Problem Information

Difficulty: Easy
Acceptance Rate: 61.58%
Paid Only: No
Tags: Array

Problem Description

Given an integer array `nums`, return the number of subarrays of length 3 such that the sum of the first and third numbers equals _exactly_ half of the second number.

Example 1:

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

Output: 1

Explanation:

Only the subarray `[1,4,1]` contains exactly 3 elements where the sum of the first and third numbers equals half the middle number.

Example 2:

Input: nums = [1,1,1]

Output: 0

Explanation:

`[1,1,1]` is the only subarray of length 3. However, its first and third numbers do not add to half the middle number.

Constraints:

* `3 <= nums.length <= 100` * `-100 <= nums[i] <= 100`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public int countSubarrays(int[] nums) {

}
}
```

Python3:

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