Problem 2348: Number of Zero-Filled Subarrays

Problem Information

Difficulty: Medium
Acceptance Rate: 70.18%
Paid Only: No
Tags: Array, Math

Problem Description

Given an integer array `nums`, return _the number of**subarrays** filled with _`0`.

A **subarray** is a contiguous non-empty sequence of elements within an array.

Example 1:

Input: nums = [1,3,0,0,2,0,0,4] **Output:** 6 **Explanation:** There are 4 occurrences of [0] as a subarray. There are 2 occurrences of [0,0] as a subarray. There is no occurrence of a subarray with a size more than 2 filled with 0. Therefore, we return 6.

Example 2:

Input: nums = [0,0,0,2,0,0] **Output:** 9 **Explanation:** There are 5 occurrences of [0] as a subarray. There are 3 occurrences of [0,0] as a subarray. There is 1 occurrence of [0,0,0] as a subarray. There is no occurrence of a subarray with a size more than 3 filled with 0. Therefore, we return 9.

Example 3:

Input: nums = [2,10,2019] **Output:** 0 **Explanation:** There is no subarray filled with 0. Therefore, we return 0.

Constraints:

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

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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