Problem 2750: Ways to Split Array Into Good Subarrays

Problem Information

Difficulty: Medium
Acceptance Rate: 34.51%
Paid Only: No
Tags: Array, Math, Dynamic Programming

Problem Description

You are given a binary array `nums`.

A subarray of an array is **good** if it contains **exactly** **one** element with the value `1`.

Return _an integer denoting the number of ways to split the array_`nums` _into**good** subarrays_. As the number may be too large, return it **modulo** `109 + 7`.

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

Example 1:

Input: nums = [0,1,0,0,1] **Output:** 3 **Explanation:** There are 3 ways to split nums into good subarrays: - [0,1] [0,0,1] - [0,1,0] [0,1] - [0,1,0,0] [1]

Example 2:

Input: nums = [0,1,0] **Output:** 1 **Explanation:** There is 1 way to split nums into good subarrays: - [0,1,0]

Constraints:

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

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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