Problem 2270: Number of Ways to Split Array

Problem Information

Difficulty: Medium
Acceptance Rate: 55.97%
Paid Only: No
Tags: Array, Prefix Sum

Problem Description

You are given a **0-indexed** integer array `nums` of length `n`.

`nums` contains a **valid split** at index `i` if the following are true:

* The sum of the first `i + 1` elements is **greater than or equal to** the sum of the last `n - i - 1` elements. * There is **at least one** element to the right of `i`. That is, `0 <= i < n - 1`.

Return _the number of**valid splits** in_ `nums`.

Example 1:

Input: nums = [10,4,-8,7] **Output:** 2 **Explanation:** There are three ways of splitting nums into two non-empty parts: - Split nums at index 0. Then, the first part is [10], and its sum is 10. The second part is [4,-8,7], and its sum is 3. Since 10 >= 3, i = 0 is a valid split. - Split nums at index 1. Then, the first part is [10,4], and its sum is 14. The second part is [-8,7], and its sum is -1. Since 14 >= -1, i = 1 is a valid split. - Split nums at index 2. Then, the first part is [10,4,-8], and its sum is 6. The second part is [7], and its sum is 7. Since 6 < 7, i = 2 is not a valid split. Thus, the number of valid splits in nums is 2.

Example 2:

Input: nums = [2,3,1,0] **Output:** 2 **Explanation:** There are two valid splits in nums: - Split nums at index 1. Then, the first part is [2,3], and its sum is 5. The second part is [1,0], and its sum is 1. Since 5 >= 1, i = 1 is a valid split. - Split nums at index 2. Then, the first part is [2,3,1], and its sum is 6. The second part is [0], and its sum is 0. Since 6 >= 0, i = 2 is a valid split.

Constraints:

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

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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