Problem 3427: Sum of Variable Length Subarrays

Problem Information

Difficulty: Easy
Acceptance Rate: 85.31%
Paid Only: No
Tags: Array, Prefix Sum

Problem Description

You are given an integer array `nums` of size `n`. For **each** index `i` where `0 <= i < n`, define a subarray `nums[start ... i]` where `start = max(0, i - nums[i])`.

Return the total sum of all elements from the subarray defined for each index in the array.

Example 1:

Input: nums = [2,3,1]

Output: 11

Explanation:

i | Subarray | Sum ---|---|--- 0 | `nums[0] = [2]` | 2 1 | `nums[0 ... 1] = [2, 3]` | 5 2 | `nums[1 ... 2] = [3, 1]` | 4 **Total Sum** | | 11 The total sum is 11. Hence, 11 is the output.

Example 2:

Input: nums = [3,1,1,2]

Output: 13

Explanation:

i | Subarray | Sum ---|---|--- 0 | `nums[0] = [3]` | 3 1 | `nums[0 ... 1] = [3, 1]` | 4 2 | `nums[1 ... 2] = [1, 1]` | 2 3 | `nums[1 ... 3] = [1, 1, 2]` | 4 **Total Sum** | | 13 The total sum is 13. Hence, 13 is the output.

Constraints:

* `1 <= n == nums.length <= 100` * `1 <= nums[i] <= 1000`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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