Problem 2996: Smallest Missing Integer Greater Than Sequential Prefix Sum

Problem Information

Difficulty: Easy
Acceptance Rate: 34.70%
Paid Only: No
Tags: Array, Hash Table, Sorting

Problem Description

You are given a **0-indexed** array of integers `nums`.

A prefix `nums[0..i]` is **sequential** if, for all `1 <= j <= i`, `nums[j] = nums[j - 1] + 1`. In particular, the prefix consisting only of `nums[0]` is **sequential**.

Return _the**smallest** integer_ `x` _missing from_ `nums` _such that_ `x` _is greater than or equal to the sum of the**longest** sequential prefix._

Example 1:

Input: nums = [1,2,3,2,5] **Output:** 6 **Explanation:** The longest sequential prefix of nums is [1,2,3] with a sum of 6. 6 is not in the array, therefore 6 is the smallest missing integer greater than or equal to the sum of the longest sequential prefix.

Example 2:

Input: nums = [3,4,5,1,12,14,13] **Output:** 15 **Explanation:** The longest sequential prefix of nums is [3,4,5] with a sum of 12. 12, 13, and 14 belong to the array while 15 does not. Therefore 15 is the smallest missing integer greater than or equal to the sum of the longest sequential prefix.

Constraints:

* `1 <= nums.length <= 50` * `1 <= nums[i] <= 50`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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