Problem 2439: Minimize Maximum of Array

Problem Information

Difficulty: Medium
Acceptance Rate: 46.45%
Paid Only: No
Tags: Array, Binary Search, Dynamic Programming, Greedy, Prefix Sum

Problem Description

You are given a **0-indexed** array `nums` comprising of `n` non-negative integers.

In one operation, you must:

* Choose an integer `i` such that `1 <= i < n` and `nums[i] > 0`. * Decrease `nums[i]` by 1. * Increase `nums[i - 1]` by 1.

Return _the**minimum** possible value of the **maximum** integer of _`nums` _after performing**any** number of operations_.

Example 1:

Input: nums = [3,7,1,6] **Output:** 5 **Explanation:** One set of optimal operations is as follows: 1. Choose i = 1, and nums becomes [4,6,1,6]. 2. Choose i = 3, and nums becomes [4,6,2,5]. 3. Choose i = 1, and nums becomes [5,5,2,5]. The maximum integer of nums is 5. It can be shown that the maximum number cannot be less than 5. Therefore, we return 5.

Example 2:

Input: nums = [10,1] **Output:** 10 **Explanation:** It is optimal to leave nums as is, and since 10 is the maximum value, we return 10.

Constraints:

* `n == nums.length` * `2 <= n <= 105` * `0 <= nums[i] <= 109`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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