Problem 1827: Minimum Operations to Make the Array Increasing

Problem Information

Difficulty: Easy
Acceptance Rate: 81.60%
Paid Only: No
Tags: Array, Greedy

Problem Description

You are given an integer array `nums` (**0-indexed**). In one operation, you can choose an element of the array and increment it by `1`.

* For example, if `nums = [1,2,3]`, you can choose to increment `nums[1]` to make `nums = [1,_**3**_ ,3]`.

Return _the**minimum** number of operations needed to make_ `nums` _**strictly** **increasing**._

An array `nums` is **strictly increasing** if `nums[i] < nums[i+1]` for all `0 <= i < nums.length - 1`. An array of length `1` is trivially strictly increasing.

Example 1:

Input: nums = [1,1,1] **Output:** 3 **Explanation:** You can do the following operations: 1) Increment nums[2], so nums becomes [1,1,_**2**_]. 2) Increment nums[1], so nums becomes [1,_**2**_ ,2]. 3) Increment nums[2], so nums becomes [1,2,_**3**_].

Example 2:

Input: nums = [1,5,2,4,1] **Output:** 14

Example 3:

Input: nums = [8] **Output:** 0

Constraints:

* `1 <= nums.length <= 5000` * `1 <= nums[i] <= 104`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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