Problem 2289: Steps to Make Array Non-decreasing

Problem Information

Difficulty: Medium
Acceptance Rate: 23.69%
Paid Only: No
Tags: Array, Linked List, Stack, Monotonic Stack

Problem Description

You are given a **0-indexed** integer array `nums`. In one step, **remove** all elements `nums[i]` where `nums[i - 1] > nums[i]` for all `0 < i < nums.length`.

Return _the number of steps performed until_`nums` _becomes a**non- decreasing** array_.

Example 1:

Input: nums = [5,3,4,4,7,3,6,11,8,5,11] **Output:** 3 **Explanation:** The following are the steps performed: - Step 1: [5,**_3_** ,4,4,7,_**3**_ ,6,11,_**8**_ ,_**5**_ ,11] becomes [5,4,4,7,6,11,11] - Step 2: [5,_**4**_ ,4,7,_**6**_ ,11,11] becomes [5,4,7,11,11] - Step 3: [5,_**4**_ ,7,11,11] becomes [5,7,11,11] [5,7,11,11] is a non-decreasing array. Therefore, we return 3.

Example 2:

Input: nums = [4,5,7,7,13] **Output:** 0 **Explanation:** nums is already a non-decreasing array. Therefore, we return 0.

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 109`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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