Problem 2972: Count the Number of Incremovable Subarrays II

Problem Information

Difficulty: Hard
Acceptance Rate: 39.77%
Paid Only: No
Tags: Array, Two Pointers, Binary Search

Problem Description

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

A subarray of `nums` is called **incremovable** if `nums` becomes **strictly increasing** on removing the subarray. For example, the subarray `[3, 4]` is an incremovable subarray of `[5, 3, 4, 6, 7]` because removing this subarray changes the array `[5, 3, 4, 6, 7]` to `[5, 6, 7]` which is strictly increasing.

Return _the total number of**incremovable** subarrays of_ `nums`.

Note that an empty array is considered strictly increasing.

A **subarray** is a contiguous non-empty sequence of elements within an array.

Example 1:

Input: nums = [1,2,3,4] **Output:** 10 **Explanation:** The 10 incremovable subarrays are: [1], [2], [3], [4], [1,2], [2,3], [3,4], [1,2,3], [2,3,4], and [1,2,3,4], because on removing any one of these subarrays nums becomes strictly increasing. Note that you cannot select an empty subarray.

Example 2:

Input: nums = [6,5,7,8] **Output:** 7 **Explanation:** The 7 incremovable subarrays are: [5], [6], [5,7], [6,5], [5,7,8], [6,5,7] and [6,5,7,8]. It can be shown that there are only 7 incremovable subarrays in nums.

Example 3:

Input: nums = [8,7,6,6] **Output:** 3 **Explanation:** The 3 incremovable subarrays are: [8,7,6], [7,6,6], and [8,7,6,6]. Note that [8,7] is not an incremovable subarray because after removing [8,7] nums becomes [6,6], which is sorted in ascending order but not strictly increasing.

Constraints:

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

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public long incremovableSubarrayCount(int[] nums) {

}
}
```

Python3:

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