Problem 1574: Shortest Subarray to be Removed to Make Array Sorted

Problem Information

Difficulty: Medium
Acceptance Rate: 51.32%
Paid Only: No
Tags: Array, Two Pointers, Binary Search, Stack, Monotonic Stack

Problem Description

Given an integer array `arr`, remove a subarray (can be empty) from `arr` such that the remaining elements in `arr` are **non-decreasing**.

Return _the length of the shortest subarray to remove_.

A **subarray** is a contiguous subsequence of the array.

Example 1:

Input: arr = [1,2,3,10,4,2,3,5] **Output:** 3 **Explanation:** The shortest subarray we can remove is [10,4,2] of length 3. The remaining elements after that will be [1,2,3,3,5] which are sorted. Another correct solution is to remove the subarray [3,10,4].

Example 2:

Input: arr = [5,4,3,2,1] **Output:** 4 **Explanation:** Since the array is strictly decreasing, we can only keep a single element. Therefore we need to remove a subarray of length 4, either [5,4,3,2] or [4,3,2,1].

Example 3:

Input: arr = [1,2,3] **Output:** 0 **Explanation:** The array is already non-decreasing. We do not need to remove any elements.

Constraints:

* `1 <= arr.length <= 105` * `0 <= arr[i] <= 109`

Code Snippets

C++:

```
class Solution {
public:
int findLengthOfShortestSubarray(vector<int>& arr) {

}
};
```

Java:

```
class Solution {
public int findLengthOfShortestSubarray(int[] arr) {

}
}
```

Python3:

```
class Solution:
def findLengthOfShortestSubarray(self, arr: List[int]) -> int:
```