Problem 2334: Subarray With Elements Greater Than Varying Threshold

Problem Information

Difficulty: Hard
Acceptance Rate: 45.00%
Paid Only: No
Tags: Array, Stack, Union Find, Monotonic Stack

Problem Description

You are given an integer array `nums` and an integer `threshold`.

Find any subarray of `nums` of length `k` such that **every** element in the subarray is **greater** than `threshold / k`.

Return _the**size** of **any** such subarray_. If there is no such subarray, return `-1`.

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

Example 1:

Input: nums = [1,3,4,3,1], threshold = 6 **Output:** 3 **Explanation:** The subarray [3,4,3] has a size of 3, and every element is greater than 6 / 3 = 2. Note that this is the only valid subarray.

Example 2:

Input: nums = [6,5,6,5,8], threshold = 7 **Output:** 1 **Explanation:** The subarray [8] has a size of 1, and 8 > 7 / 1 = 7. So 1 is returned. Note that the subarray [6,5] has a size of 2, and every element is greater than 7 / 2 = 3.5. Similarly, the subarrays [6,5,6], [6,5,6,5], [6,5,6,5,8] also satisfy the given conditions. Therefore, 2, 3, 4, or 5 may also be returned.

Constraints:

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

Code Snippets

C++:

```
class Solution {
public:
int validSubarraySize(vector<int>& nums, int threshold) {

}
};
```

Java:

```
class Solution {
public int validSubarraySize(int[] nums, int threshold) {

}
}
```

Python3:

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