Problem 3676: Count Bowl Subarrays

Problem Information

Difficulty: Medium
Acceptance Rate: 43.91%
Paid Only: No
Tags: Array, Stack, Monotonic Stack

Problem Description

You are given an integer array `nums` with **distinct** elements.

A subarray `nums[l...r]` of `nums` is called a **bowl** if:

* The subarray has length at least 3. That is, `r - l + 1 >= 3`. * The **minimum** of its two ends is **strictly greater** than the **maximum** of all elements in between. That is, `min(nums[l], nums[r]) > max(nums[l + 1], ..., nums[r - 1])`.

Return the number of **bowl** subarrays in `nums`.

Example 1:

Input: nums = [2,5,3,1,4]

Output: 2

Explanation:

The bowl subarrays are `[3, 1, 4]` and `[5, 3, 1, 4]`.

* `[3, 1, 4]` is a bowl because `min(3, 4) = 3 > max(1) = 1`. * `[5, 3, 1, 4]` is a bowl because `min(5, 4) = 4 > max(3, 1) = 3`.

Example 2:

Input: nums = [5,1,2,3,4]

Output: 3

Explanation:

The bowl subarrays are `[5, 1, 2]`, `[5, 1, 2, 3]` and `[5, 1, 2, 3, 4]`.

Example 3:

Input: nums = [1000000000,999999999,999999998]

Output: 0

Explanation:

No subarray is a bowl.

Constraints:

* `3 <= nums.length <= 105` * `1 <= nums[i] <= 109` * `nums` consists of distinct elements.

Code Snippets

C++:

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

}
};
```

Java:

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

}
```

```
}
```

Python3:

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