Problem 1950: Maximum of Minimum Values in All Subarrays

Problem Information

Difficulty: Medium
Acceptance Rate: 48.17%
Paid Only: Yes
Tags: Array, Stack, Monotonic Stack

Problem Description

You are given an integer array `nums` of size `n`. You are asked to solve `n` queries for each integer `i` in the range `0 <= i < n`.

To solve the `ith` query:

1. Find the **minimum value** in each possible subarray of size `i + 1` of the array `nums`. 2. Find the **maximum** of those minimum values. This maximum is the **answer** to the query.

Return _a**0-indexed** integer array_ `ans` _of size_`n` _such that_`ans[i]` _is the answer to the_`ith` _query_.

A **subarray** is a contiguous sequence of elements in an array.

Example 1:

Input: nums = [0,1,2,4] **Output:** [4,2,1,0] **Explanation:** i=0: - The subarrays of size 1 are [0], [1], [2], [4]. The minimum values are 0, 1, 2, 4. - The maximum of the minimum values is 4. i=1: - The subarrays of size 2 are [0,1], [1,2], [2,4]. The minimum values are 0, 1, 2. - The maximum of the minimum values is 2. i=2: - The subarrays of size 3 are [0,1,2], [1,2,4]. The minimum values are 0, 1. - The maximum of the minimum values is 1. i=3: - There is one subarray of size 4, which is [0,1,2,4]. The minimum value is 0. - There is only one value, so the maximum is 0.

Example 2:

Input: nums = [10,20,50,10] **Output:** [50,20,10,10] **Explanation:** i=0: - The subarrays of size 1 are [10], [20], [50], [10]. The minimum values are 10, 20, 50, 10. - The maximum of the minimum values is 50. i=1: - The subarrays of size 2 are [10,20], [20,50], [50,10]. The minimum values are 10, 20, 10. - The maximum of the minimum values is 20. i=2: - The subarrays of size 3 are [10,20,50], [20,50,10]. The minimum values are 10, 10. - The maximum of the minimum values is 10. i=3: - There is one subarray of size 4, which is [10,20,50,10]. The minimum value is 10. - There is only one value, so the maximum is 10.

Constraints:

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

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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