Problem 3640: Trionic Array II

Problem Information

Difficulty: Hard
Acceptance Rate: 24.71%
Paid Only: No
Tags: Array, Dynamic Programming

Problem Description

You are given an integer array `nums` of length `n`.

A **trionic subarray** is a contiguous subarray `nums[l...r]` (with `0 <= l < r < n`) for which there exist indices `l < p < q < r` such that:

* `nums[l...p]` is **strictly** increasing, * `nums[p...q]` is **strictly** decreasing, * `nums[q...r]` is **strictly** increasing.

Return the **maximum** sum of any trionic subarray in `nums`.

Example 1:

Input: nums = [0,-2,-1,-3,0,2,-1]

Output: -4

Explanation:

Pick `l = 1`, `p = 2`, `q = 3`, `r = 5`:

* `nums[l...p] = nums[1...2] = [-2, -1]` is strictly increasing (`-2 < -1`). * `nums[p...q] = nums[2...3] = [-1, -3]` is strictly decreasing (`-1 > -3`) * `nums[q...r] = nums[3...5] = [-3, 0, 2]` is strictly increasing (`-3 < 0 < 2`). * Sum = `(-2) + (-1) + (-3) + 0 + 2 = -4`.

Example 2:

Input: nums = [1,4,2,7]

Output: 14

Explanation:

Pick `l = 0`, `p = 1`, `q = 2`, `r = 3`:

* `nums[l...p] = nums[0...1] = [1, 4]` is strictly increasing (`1 < 4`). * `nums[p...q] = nums[1...2] = [4, 2]` is strictly decreasing (`4 > 2`). * `nums[q...r] = nums[2...3] = [2, 7]` is strictly increasing (`2 < 7`). * Sum = `1 + 4 + 2 + 7 = 14`.

Constraints:

* `4 <= n = nums.length <= 105` * `-109 <= nums[i] <= 109` * It is guaranteed that at least one trionic subarray exists.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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