Problem 3637: Trionic Array I

Problem Information

Difficulty: Easy
Acceptance Rate: 44.85%
Paid Only: No
Tags: Array

Problem Description

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

An array is **trionic** if there exist indices `0 < p < q < n – 1` such that:

* `nums[0...p]` is **strictly** increasing, * `nums[p...q]` is **strictly** decreasing, * `nums[q...n – 1]` is **strictly** increasing.

Return `true` if `nums` is trionic, otherwise return `false`.

Example 1:

Input: nums = [1,3,5,4,2,6]

Output: true

Explanation:

Pick `p = 2`, `q = 4`:

* `nums[0...2] = [1, 3, 5]` is strictly increasing (`1 < 3 < 5`). * `nums[2...4] = [5, 4, 2]` is strictly decreasing (`5 > 4 > 2`). * `nums[4...5] = [2, 6]` is strictly increasing (`2 < 6`).

Example 2:

Input: nums = [2,1,3]

Output: false

Explanation:

There is no way to pick `p` and `q` to form the required three segments.

Constraints:

* `3 <= n <= 100` * `-1000 <= nums[i] <= 1000`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public boolean isTrionic(int[] nums) {

}
}
```

Python3:

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