Problem 456: 132 Pattern

Problem Information

Difficulty: Medium
Acceptance Rate: 34.36%
Paid Only: No
Tags: Array, Binary Search, Stack, Monotonic Stack, Ordered Set

Problem Description

Given an array of `n` integers `nums`, a **132 pattern** is a subsequence of three integers `nums[i]`, `nums[j]` and `nums[k]` such that `i < j < k` and `nums[i] < nums[k] < nums[j]`.

Return `true` _if there is a**132 pattern** in _`nums` _, otherwise, return_`false` _._

Example 1:

Input: nums = [1,2,3,4] **Output:** false **Explanation:** There is no 132 pattern in the sequence.

Example 2:

Input: nums = [3,1,4,2] **Output:** true **Explanation:** There is a 132 pattern in the sequence: [1, 4, 2].

Example 3:

Input: nums = [-1,3,2,0] **Output:** true **Explanation:** There are three 132 patterns in the sequence: [-1, 3, 2], [-1, 3, 0] and [-1, 2, 0].

Constraints:

* `n == nums.length` * `1 <= n <= 2 * 105` * `-109 <= nums[i] <= 109`

Code Snippets

C++:

```
class Solution {
public:
bool find132pattern(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public boolean find132pattern(int[] nums) {

}
}
```

Python3:

```
class Solution:
def find132pattern(self, nums: List[int]) -> bool:
```