456. 132 Pattern

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 * 10<sup>5</sup>`
- `-10<sup>9</sup> <= nums[i] <= 10<sup>9</sup>`

```
class Solution:
    def find132pattern(self, nums: List[int]) -> bool:
        minimum = [0]*len(nums)

        minimum[0] = nums[0]
        for i in range(1,len(nums)):
            minimum[i] = min(minimum[i-1],nums[i])

        stack = []
        for i in range(len(nums)-1,-1,-1):
            while len(stack) and stack[-1]<=minimum[i]:
                stack.pop()
            if len(stack) and stack[-1]<nums[i]:
                return True
            stack.append(nums[i])
        return False
```