456. 132 Pattern

Add to List
QuestionEditorial Solution
My Submissions

- Total Accepted: **3145**
- Total Submissions: **11701**
- Difficulty: **Medium**
- Contributors: **love_FDU_llp**

Given a sequence of n integers a_1, a_2, ..., a_n, a 132 pattern is a subsequence a_i, a_j, a_k such that **i** < **j** < **k** and $a_i < a_k < a_j$. Design an algorithm that takes a list of n numbers as input and checks whether there is a 132 pattern in the list.

Note: n will be less than 15,000.

Example 1:

```
Input: [1, 2, 3, 4]

Output: False

Explanation: There is no 132 pattern in the sequence.
```

Example 2:

```
Input: [3, 1, 4, 2]

Output: True

Explanation: There is a 132 pattern in the sequence: [1, 4, 2].
```

Example 3:

```
Input: [-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].
```

MPLEMENTATION:

1. Have a stack, each time we store a new number, we first pop out all numbers that are smaller than that number. The numbers that are popped out becomes candidate for s3.

2. We keep track of the maximum of such s3 (which is always the most recently popped number from the stack).
3. Once we encounter any number smaller than s3, we know we found a valid sequence since s1 < s3 implies s1 < s2.

RUNTIME: Each item is pushed and popped once at most, the time complexity is therefore O(n).
EXAMPLE:
i = 6, **nums** = [9, 11, 8, 9, 10, 7, 9], **S1 candidate** = 9, **S3 candidate** = None, **Stack** = Empty
i = 5, **nums** = [9, 11, 8, 9, 10, 7, 9], **S1 candidate** = 7, **S3 candidate** = None, **Stack** = [9]
i = 4, **nums** = [9, 11, 8, 9, 10, 7, 9], **S1 candidate** = 10, **S3 candidate** = None, **Stack** = [9,7]
i = 3, **nums** = [9, 11, 8, 9, 10, 7, 9], **S1 candidate** = 9, **S3 candidate** = 9, **Stack** = [10]
i = 2, **nums** = [9, 11, 8, 9, 10, 7, 9], **S1 candidate** = 8, **S3 candidate** = 9, **Stack** = [10,9] **We have 8<9, sequence found!**

```
class Solution {
public:
    bool find132pattern(vector<int>& nums) {
        stack<int> st;
        int s3 = INT_MIN;
        for(int i=nums.size()-1;i>=0;i--)
        {
            if(nums[i]<s3) return true;
            else while(!st.empty() && nums[i]>st.top())
            {
                s3 = st.top();st.pop();
            }
            st.push(nums[i]);
        }
        return false;
    }
};
```