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

public class Solution {

public boolean find132pattern(int[] nums) {

Stack<Integer> stack = new Stack<>();

int third = Integer.MIN\_VALUE;

for (int i = nums.length - 1; i >= 0; i--) {

if (nums[i] < third) return true;

while (!stack.isEmpty() && stack.peek() < nums[i]) {

third = stack.pop();

}

stack.push(nums[i]);

}

return false;

}

}