Valid Mountain Array

# Question

Given an array of integers arr, return true if and only if it is a valid mountain array.

Recall that arr is a mountain array if and only if:

arr.length >= 3

There exists some i with 0 < i < arr.length - 1 such that:

arr[0] < arr[1] < ... < arr[i - 1] < arr[i]

arr[i] > arr[i + 1] > ... > arr[arr.length - 1]

**Example 1:**

Input: arr = [2,1]

Output: false

**Example 2:**

Input: arr = [3,5,5]

Output: false

**Example 3:**

Input: arr = [0,3,2,1]

Output: true

# Pseudo Code

If size of the array < 3

Return False

Find the Index of the Largest Element in the Array

Run the For Loop

Check whether the Elements on the left side of that element are strictly increasing and that on the right side are strictly decreasing

If so

Return True

Else

Return False

Return False

# Source Code

## V1.0 (Language: C)

1. bool validMountainArray(int\* arr, int arrSize){
3. if (arrSize < 3) {
4. return false;
5. }
7. int maxElement = 0, maxElementIndex = 0;
8. for(int i=0 ; i<arrSize ; i++) {
10. if(maxElement < arr[i]) {
11. maxElement = arr[i];
12. maxElementIndex = i;
13. }
14. }
16. if (maxElementIndex == 0 || maxElementIndex == (arrSize - 1)) {
17. return false;
18. }
20. for(int i=0 ; i<maxElementIndex ; i++) {
21. if(arr[i + 1] <= arr[i])
22. return false;
23. }
25. for(int i=maxElementIndex ; i<(arrSize-1) ; i++) {
26. if(arr[i + 1] >= arr[i])
27. return false;
28. }
29. return true;
30. }