DSA

1st Array problems

Searching :

1. **You are given an integer mountain array arr of length n where the values increase to a peak element and then decrease. Return the index of the peak element. Your task is to solve it in O(log(n)) time complexity.**

Example 1:

Input: arr = [0,1,0]

Output: 1

Example 2:

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

Output: 1

Example 3:

Input: arr = [0,10,5,2]

Output: 1

Constraints:

- 3 <= arr.length <= 10^5
- 0 <= arr[i] <= 10^6
- arr is guaranteed to be a mountain array.

- A peak element in an array is defined as an element that is greater than or equal to its adjacent elements. More formally, for an element arr[i] to be a peak element, it must satisfy the condition arr[i] >= arr[i-1] and arr[i] >= arr[i+1]. For edge cases, if the element is at the beginning or end of the array, it only needs to be greater than its single neighbor. It's also important to note that an array can have multiple peak elements.

Input: arr[] = [1, 2, 4, 5, 7, 8, 3]
Output: 5
Explanation: arr[5] = 8 is a peak element because arr[4] < arr[5] > arr[6].

Input: arr[] = [10, 20, 15, 2, 23, 90, 80]
Output: 1 or 5
Explanation: arr[1] = 20 and arr[5] = 90 are peak elements because arr[0] < arr[1] > arr[2] and arr[4] < arr[5] > arr[6].

Input: arr[] = [1, 2, 3]
Output: 2
Explanation: arr[2] is a peak element because arr[1] < arr[2] and arr[2] is the last element, so it has negative infinity to its right.