

852. Peak Index in a Mountain Array

Easy

👍 1167

💬 1393

❤ Add to List

📄 Share

Let's call an array `arr` a **mountain** if the following properties hold:

- `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]`

Given an integer array `arr` that is **guaranteed** to be a mountain, return any `i` such that `arr[0] < arr[1] < ... arr[i - 1] < arr[i] > arr[i + 1] > ... > arr[arr.length - 1]`.

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

Example 4:

Input: arr = [3,4,5,1]

Output: 2

Example 5:

Input: arr = [24,69,100,99,79,78,67,36,26,19]

Output: 2

Constraints:

- `3 <= arr.length <= 104`
- `0 <= arr[i] <= 106`
- `arr` is **guaranteed** to be a mountain array.

Follow up: Finding the `O(n)` is straightforward, could you find an `O(log(n))` solution?

Related Topics



Binary Search

Similar Questions



Find Peak Element

Medium

Find in Mountain Array

Hard

Minimum Number of Removals to Make Mountain Array

Hard