https://course.acciojob.com/idle?question=fc5f657f-bf71-49c6-890c-96a5835f684b

- EASY
- Max Score: 30 Points
-
-
-
-

a

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

Given a mountain array arr, return the index i such that arr[0] < arr[1] < ... < arr[i - 1] < arr[i] > arr[i + 1] > ... > arr[arr.length - 1].

You must solve it in O(log(arr.length)) time complexity.

Input Format

Input consists of two lines.

First line contains an integer `n`.

Next line contains `n` spaced integers.

Output Format

Output is the index of the peak element.

Example 1

Input

```
3
0 1 0
```

Output

```
1
```

Example 2

Input

```
4
0 2 1 0
```

Output

```
1
```

Example 3

Input

```
4
0 10 5 2
```

Output

```
1
```

Constraints

`3 <= arr.length <= 105`

`0 <= arr[i] <= 106`

`arr` is guaranteed to be a mountain array.

Topic Tags

- Binary Search
- Arrays

My code

```
// n java
import java.io.*;
import java.util.*;

class Solution {
    public int peakIndexInMountainArray(int[] arr) {
      // Your code here.
          int n=arr.length;
          int ans=-1;
          int lp=0,rp=n-1;
          while(lp<=rp)
                {
                      int mid=(lp+rp)/2;
                      if(arr[mid]>arr[mid+1] && arr[mid]>arr[mid-1])
                      {
                            ans=mid;
                            break;
                      }

                        if(arr[mid]>arr[mid-1])
                            lp=mid+1;
                      else rp=mid-1;
                }
          return ans;
```

```
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n;
        n = sc.nextInt();
        int arr1[] = new int[n];
        for(int i=0;i<n;i++)
            arr1[i] = sc.nextInt();
        Solution Obj = new Solution();
        int result = Obj.peakIndexInMountainArray(arr1);
        System.out.print(result + " ");
        System.out.println('\n');
    }
}
```