162. Find Peak Element

A peak element is an element that is strictly greater than its neighbors.

Given an integer array `nums`, find a peak element, and return its index. If the array contains multiple peaks, return the index to **any of the peaks**.

You may imagine that `nums[-1] = nums[n] = -∞`.

You must write an algorithm that runs in `O(log n)` time.

Example 1:

Input: nums = [1,2,3,1]
Output: 2
Explanation: 3 is a peak element and your function should return the index number 2.

Example 2:

Input: nums = [1,2,1,3,5,6,4]
Output: 5
Explanation: Your function can return either index number 1 where the peak element is 2,
or index number 5 where the peak element is 6.

```
def findPeakElement(self, arr: List[int]) -> int:
        if len(arr)==1:
            return 0
        # lo = 0
        # n = len(arr)
        # hi = n-1
        # while lo<=hi:
        #     mid = lo+(hi-lo)//2
        #     prev = (mid-1+n)%n
        #     nextt = (mid+1+n)%n
        #     if mid>0 and mid<n-1:
        #         if arr[mid] > arr[nextt] and arr[mid] > arr[prev]:
        #             return mid
        #         elif arr[mid] > arr[mid+1]:
        #             hi = mid-1
        #         elif arr[mid] < arr[mid+1]:
        #             lo = mid +1
        #     elif mid==0:
```

```
        #         if arr[mid]>arr[1]:
        #             return 0
        #         else:
        #             return 1
        #     elif mid==n-1:
        #         if arr[n-1]>arr[n-2]:
        #             return n-1
        #         else:
        #             return n-2

        #Best Approach
        lo, hi = 0, len(arr) - 1
        while lo < hi:
            mi = (lo + hi) // 2
            if arr[mi] < arr[mi + 1]:
                lo = mi + 1
            else:
                hi = mi
        return lo
```