

147. A peak element is an element that is strictly greater than its neighbors. Given a 0-indexed 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] = -∞`. In other words, an element is always considered to be strictly greater than a neighbor that is outside the array. You must write an algorithm that runs in $O(\log n)$ time.

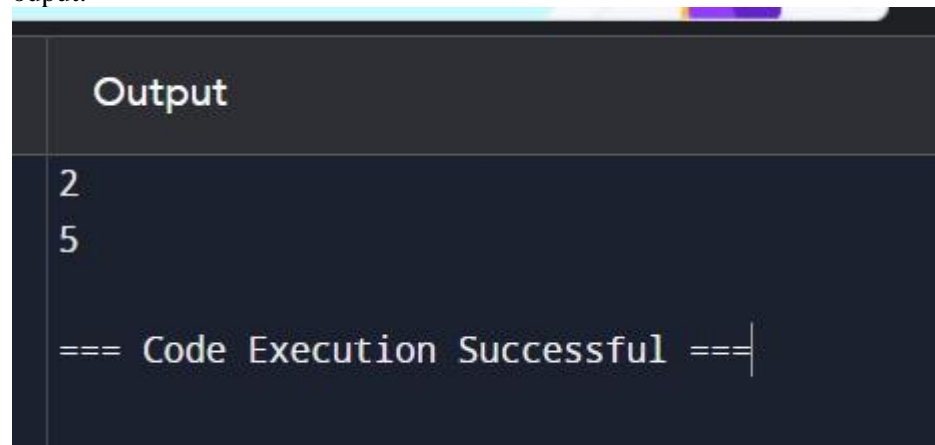
Program:-

```
def find_peak_element(nums):  
    left, right = 0, len(nums) - 1  
  
    while left < right:  
        mid = left + (right - left) // 2  
  
        if nums[mid] < nums[mid + 1]:  
            left = mid + 1  
        else:  
            right = mid  
  
    return left
```

input:-

`nums1 = [1, 2, 3, 1]`

output:-

A screenshot of a code execution environment. It features a dark-themed window with a title bar. The window contains the text "Output" in a light font. Below this, the numbers "2" and "5" are displayed on separate lines. At the bottom of the window, a message "=== Code Execution Successful ===" is shown, with a vertical cursor at the end of the line.

```
Output  
2  
5  
  
=== Code Execution Successful ===
```

Time complexity:- $O(\log n)$