0162-Find Peak Element

Description

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

Given an input array nums, where nums[i] ≠ nums[i+1], find a peak element and return its index.

The array may contain multiple peaks, in that case return the index to any one of the peaks is fine.

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

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: 1 or 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.

Note:

Your solution should be in logarithmic complexity.

Solution

- 一定要注意讨论切中valley的情况，非常容易忽略。比如：[2, 1, 2]这种case。
- 时间复杂度：O(logN)
- 空间复杂度：O(1)

```
class Solution {
    public int findPeakElement(int[] nums) {
        if(nums == null || nums.length == 0) {
            return -1;
        }
        if(nums.length == 1) {
            return 0;
        }
        // 二分法
        int start = 0;
        int end = nums.length - 1;
        while(start + 1 < end) {
            int mid = start + (end - start) / 2;
            // 切中peak
```

```
            if(nums[mid] > nums[mid-1] && nums[mid] > nums[mid+1]) {
                return mid;
            }
            // 切中上坡
            if(nums[mid] > nums[mid-1] && nums[mid] < nums[mid+1]) {
                start = mid;
            }
            // 切中下坡 或者 切中valley
            if(nums[mid] < nums[mid-1]) {
                end = mid;
            }
        }
        if(nums[start] > nums[end]) {
            return start;
        }
        return end;
    }
}
```