

0162. 寻找峰值

👤 ITCharge 🕒 大约 2 分钟

- 标签：数组、二分查找
- 难度：中等

题目链接

- [0162. 寻找峰值 - 力扣](#)

题目大意

描述： 给定一个整数数组 `nums` 。

要求： 找到峰值元素并返回其索引。必须实现时间复杂度为 $O(\log n)$ 的算法来解决此问题。

说明：

- **峰值元素：** 指其值严格大于左右相邻值的元素。
- 数组可能包含多个峰值，在这种情况下，返回任何一个峰值所在位置即可。
- 可以假设 $nums[-1] = nums[n] = -\infty$ 。
- $1 \leq nums.length \leq 1000$ 。
- $-2^{31} \leq nums[i] \leq 2^{31} - 1$ 。
- 对于所有有效的 i 都有 $nums[i] \neq nums[i + 1]$ 。

示例：

- 示例 1:

输入: `nums = [1,2,3,1]`

输出: `2`

解释: `3` 是峰值元素，你的函数应该返回其索引 `2`。

py

- 示例 2:

输入: `nums = [1,2,1,3,5,6,4]`

输出: 1 或 5

解释: 你的函数可以返回索引 1, 其峰值元素为 2; 或者返回索引 5, 其峰值元素为 6。

解题思路

思路 1: 二分查找

1. 使用两个指针 `left`、`right`。`left` 指向数组第一个元素, `right` 指向数组最后一个元素。
2. 取区间中间节点 `mid`, 并比较 `nums[mid]` 和 `nums[mid + 1]` 的值大小。
 1. 如果 `nums[mid]` 小于 `nums[mid + 1]`, 则右侧存在峰值, 令 `left = mid + 1`。
 2. 如果 `nums[mid]` 大于等于 `nums[mid + 1]`, 则左侧存在峰值, 令 `right = mid`。
3. 最后, 当 `left == right` 时, 跳出循环, 返回 `left`。

思路 1: 代码

```
class Solution:
    def findPeakElement(self, nums: List[int]) -> int:
        left = 0
        right = 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
```

思路 1: 复杂度分析

- 时间复杂度: $O(\log_2 n)$ 。
- 空间复杂度: $O(1)$ 。

