

0034. 在排序数组中查找元素的第一个和最后一个位置

👤 [ITCharge](#) ⌚ 大约 1 分钟

- 标签：数组、二分查找
- 难度：中等

题目链接

- [0034. 在排序数组中查找元素的第一个和最后一个位置 - 力扣](#)

题目大意

描述：给你一个按照非递减顺序排列的整数数组 `nums`，和一个目标值 `target`。

要求：找出给定目标值在数组中的开始位置和结束位置。

说明：

- 要求使用时间复杂度为 $O(\log n)$ 的算法解决问题。

示例：

- 示例 1：

```
输入: nums = [5,7,7,8,8,10], target = 8
输出: [3,4]
```

py

- 示例 2：

```
输入: nums = [5,7,7,8,8,10], target = 6
输出: [-1,-1]
```

py

解题思路

思路 1：二分查找

要求使用时间复杂度为 $O(\log n)$ 的算法解决问题，那么就需要使用「二分查找算法」了。

- 进行两次二分查找，第一次尽量向左搜索。第二次尽量向右搜索。

思路 1：代码

```
class Solution:
    def searchRange(self, nums: List[int], target: int) -> List[int]:
        ans = [-1, -1]
        n = len(nums)
        if n == 0:
            return ans

        left = 0
        right = n - 1
        while left < right:
            mid = left + (right - left) // 2
            if nums[mid] < target:
                left = mid + 1
            else:
                right = mid

        if nums[left] != target:
            return ans

        ans[0] = left

        left = 0
        right = n - 1
        while left < right:
            mid = left + (right - left + 1) // 2
            if nums[mid] > target:
                right = mid - 1
            else:
```

py

```
        left = mid

    if nums[left] == target:
        ans[1] = left

    return ans
```

思路 1：复杂度分析

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