说明：之前的博客电脑损坏数据丢失还没时间修复，新博客还没来得及搭建，暂时把博客内容放在这里啦。

## 704. Binary Search

```
class Solution:
        def search(self, nums: List[int], target: int) -> int:
                left, right = 0, len(nums)-1
                while left <= right:
                        middle = (left + right)//2
                        if target < nums[middle]:
                                right = middle-1
                        elif target > nums[middle]:
                                left = middle + 1
                        elif target == nums[middle]:
                                return middle
                return -1
```

注意点：

- while部分考虑两端闭合情况，所以是 <=
- 一开始脑短路写了`return nums.index(target)`，其实就是`return middle`...

相关视频与文章

## 27. Remove Element

双指针法

```
class Solution:
        def removeElement(self, nums: List[int], val: int) -> int:
                if len(nums) == 0: return 0
                l,r = 0, len(nums)-1
                while l < r:
                        while(l<r and val != nums[l]):
                                l += 1
                        while(l<r and val == nums[r]):
                                r -= 1
# remove left element covered by right element
                        nums[l], nums[r] = nums[r], nums[l]
```

```
                print(nums)
                if nums[l] == val:
                        return l
                else:
                        return l+1
```

注意点：

- 要考虑到空集的情况
- 双指针 左右交互