

704. 二分查找 + 27.移除元素

Saturday, July 2, 2022 7:40 AM

<https://leetcode.cn/problems/binary-search/submissions/>

简单

给定一个 n 个元素有序的（升序）整型数组 `nums` 和一个目标值 `target`，写一个函数搜索 `nums` 中的 `target`，如果目标值存在返回下标，否则返回 `-1`。

二分查找：（注意根据初始条件区间的开闭写对应判断条件和赋值）

1.闭区间写法 左闭右闭 `[]`，初始 `l=0,r=nums.length-1`;

while判断条件 `l<=r`

不等时赋值, `l=mid+1; r=mid-1`;

2.开区间写法 左闭右开 `[)`，初始 `l=0,r=nums.length`;

while判断条件 `l<r`

不等时赋值, `l=mid+1; r=mid`;

```
class Solution {
    public int search(int[] nums, int target) {
        if(nums==null||nums.length==0){
            return -1;
        }
        int l=0,r=nums.length-1;
        while(l<=r){
            int mid=l+(r-l)/2;
            if(target==nums[mid]){
                return mid;
            }else if(target>nums[mid]){
                l=mid+1;
            }else{
                r=mid-1;
            }
        }
        return -1;
    }
}
```

27. 移除元素

<https://leetcode.cn/problems/remove-element/>

简单

给你一个数组 `nums` 和一个值 `val`，你需要 原地 移除所有数值等于 `val` 的元素，并返回移除

后数组的新长度。

不要使用额外的数组空间，你必须仅使用 $O(1)$ 额外空间并 **原地** 修改输入数组。

示例 2:

输入: `nums = [0,1,2,2,3,0,4,2]`, `val = 2`

输出: `5`, `nums = [0,1,3,0,4]`

解法: 双指针, 快慢指针

初始化`fast`、`slow`都为0;

for循环遍历, `fast`作为l

`fast`先走, 当`nums[fast] == val` 时, 它继续走, `slow`不动

当`nums[fast] != val` 时, 将`nums[fast]`赋值给`nums[slow]`, 达到**原地**修改的效果

```
class Solution {
    public int removeElement(int[] nums, int val) {
        if(nums == null || nums.length == 0){
            return 0;
        }
        int fast=0,slow=0;
        for(fast=0;fast<nums.length;fast++){
            if(nums[fast]!=val){ // 此时为不等于目标值, 等于目标值时, fast+1即可, 直接下
                // 次循环, 无需处理
                nums[slow]=nums[fast];
                slow++;
            }
        }
        return slow;
    }
}
```