

27 移除元素

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**。

元素的顺序可以改变。你不需要考虑数组中超出新长度后面的元素。

给定 `nums = [0,1,2,2,3,0,4,2]`，`val = 2`，
函数应该返回新的长度 `5`，并且 `nums` 中的前五个元素为 `0, 1, 3, 0, 4`。
注意这五个元素可为任意顺序。

- 双指针

```
class Solution {
    public int removeElement(int[] nums, int val) {
        int replace = 0; // 一个指针控制填充的位置
        for (int i = 0; i < nums.length; i++) { // 一个指针控制循环
            if (nums[i] != val) {
                nums[replace] = nums[i];
                replace++;
            }
        }
        return replace;
    }
}
```

- 双指针——当要删除的元素很少时

```
class Solution {
    public int removeElement(int[] nums, int val) {
        int i = 0;
        int n = nums.length;
        while (i < n) {
            // 赋值操作的次数等于要删除的元素的数量。因此，如果要移除的元素很少，效率会更高。
            if (nums[i] == val) { // 将当前元素与最后一个元素进行交换，并释放最后一个元素。这实际上使数组的大小减少了 1
                nums[i] = nums[n - 1];
                // reduce array size by one
                n--;
            } else {
                i++; // 被交换的最后一个元素可能是您想要移除的值。在下一次迭代中，仍然要检查这个元素
            }
        }
        return n;
    }
}
```