

0027. 移除元素

👤 ITCharge 🕒 大约 2 分钟

- 标签：数组、双指针
- 难度：简单

题目链接

- [0027. 移除元素 - 力扣](#)

题目大意

描述： 给定一个数组 *nums*， 和一个值 *val*。

要求： 不使用额外数组空间，将数组中所有数值等于 *val* 的元素移除掉，并且返回新数组的长度。

说明：

- $0 \leq \text{nums.length} \leq 100$ 。
- $0 \leq \text{nums}[i] \leq 50$ 。
- $0 \leq \text{val} \leq 100$ 。

示例：

- 示例 1：

输入：nums = [3,2,2,3], val = 3

输出：2, nums = [2,2]

解释：函数应该返回新的长度 2，并且 nums 中的前两个元素均为 2。你不需要考虑数组中超出新长度后面的元素。例如，函数返回的新长度为 2，而 nums = [2,2,3,3] 或 nums = [2,2,0,0]，也会被视为正确答案。

py

- 示例 2：

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

输出: `5`, `nums = [0,1,4,0,3]`

解释: 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。你不需要考虑数组中超出新长度后面的元素。

解题思路

思路 1: 快慢指针

1. 使用两个指针 `slow`, `fast`。 `slow` 指向处理好的非 `val` 值元素数组的尾部, `fast` 指针指向当前待处理元素。
2. 不断向右移动 `fast` 指针, 每次移动到非 `val` 值的元素, 则将左右指针对应的数交换, 交换同时将 `slow` 右移。
3. 这样就将非 `val` 值的元素进行前移, `slow` 指针左边均为处理好的非 `val` 值元素, 而从 `slow` 指针指向的位置开始, `fast` 指针左边都为 `val` 值。
4. 遍历结束之后, 则所有 `val` 值元素都移动到了右侧, 且保持了非零数的相对位置。此时 `slow` 就是新数组的长度。

思路 1: 代码

```
class Solution:
    def removeElement(self, nums: List[int], val: int) -> int:
        slow = 0
        fast = 0
        while fast < len(nums):
            if nums[fast] != val:
                nums[slow], nums[fast] = nums[fast], nums[slow]
                slow += 1
            fast += 1
        return slow
```

思路 1: 复杂度分析

- 时间复杂度: $O(n)$ 。

- 空间复杂度: $O(1)$ 。