

切换主题：

默认主题

▼

题目地址（26.删除排序数组中的重复项）

https://leetcode-cn.com/problems/remove-duplicates-from-sorted-array/

入选理由

1. 双指针中的一种类型：读写双指针

标签

- 数组
- 双指针

难度

- 简单

题目描述

给定一个排序数组，你需要在 原地 删除重复出现的元素，使得每个元素只出现一次，返回移除后数组的新长度。

不要使用额外的数组空间，你必须在 原地 修改输入数组 并在使用 $O(1)$ 额外空间的条件下完成。

示例 1：

给定数组 `nums = [1,1,2]`，

函数应该返回新的长度 `2`，并且原数组 `nums` 的前两个元素被修改为 `1, 2`。

你不需要考虑数组中超出新长度后面的元素。

示例 2：

给定 `nums = [0,0,1,1,1,2,2,3,3,4]`，

函数应该返回新的长度 `5`，并且原数组 `nums` 的前五个元素被修改为 `0, 1, 2, 3, 4`。

你不需要考虑数组中超出新长度后面的元素。

说明：

为什么返回数值是整数，但输出的答案是数组呢？

请注意，输入数组是以「引用」方式传递的，这意味着在函数里修改输入数组对于调用者是可见的。

你可以想象内部操作如下：

```
// nums 是以“引用”方式传递的。也就是说，不对实参做任何拷贝
int len = removeDuplicates(nums);

// 在函数里修改输入数组对于调用者是可见的。
// 根据你的函数返回的长度，它会打印出数组中该长度范围内的所有元素。
for (int i = 0; i < len; i++) {
    print(nums[i]);
}
```

双指针

思路

- 用一个读指针，一个写指针遍历数组。
- 遇到重复的元素 **读指针** 就继续前移。
- 遇到不同的元素 **写指针** 就前移一步，写入那个元素。

读指针



写指针

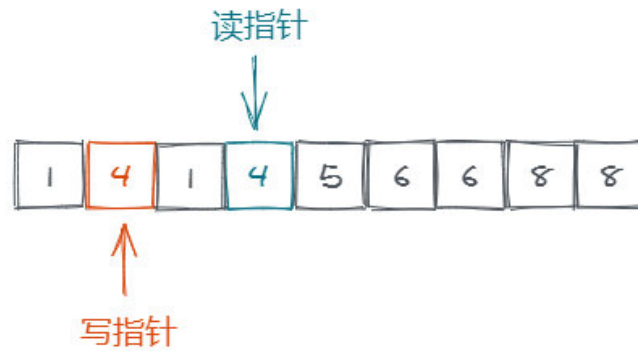
读、写指针一样时，
写指针原地不动
读指针继续往前走

读指针

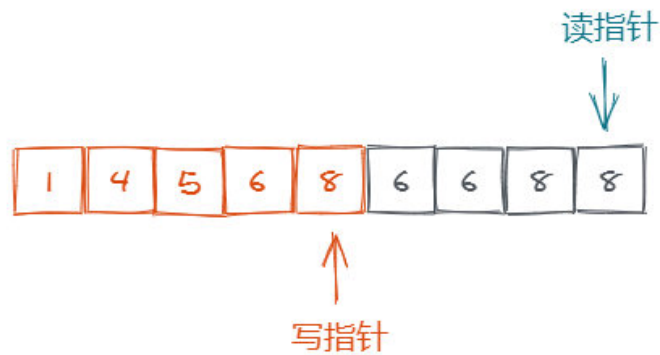


写指针

当读、写指针不同时
写指针前进一步
将读指针内容写入写指针的位置



读指针继续往前走
重复上面的步骤



最后返回写指针位置+1
就是新数组的长度了

@suvkii

代码

代码支持: JS, Python3

JS Code:

```
/**
 * @param {number[]} nums
 * @return {number}
 */
var removeDuplicates = function (nums) {
    let p1 = 0,
        p2 = 0;

    while (p2 < nums.length) {
        if (nums[p1] !== nums[p2]) {
            p1++;
            nums[p1] = nums[p2];
        }
        p2++;
    }
    return p1 + 1;
};
```

Python3 Code:

```
class Solution(object):
    def removeDuplicates(self, nums):
        """
        :type nums: List[int]
        :rtype: int
        """
        if not nums: return 0

        l, r = 0, 0
        while r < len(nums):
            if nums[l] != nums[r]:
                l += 1
                nums[l] = nums[r]
            r += 1
        return l + 1
```

复杂度分析

令 n 为数组长度

时间复杂度: $O(n)$ 空间复杂度: $O(1)$

[上一页](#)[下一页](#)

© 2020 lucifer. 保留所有权利