

画解剑指 Offer - LeetBook - 力扣 (LeetCode) 全球极客挚爱的技术成长平台

 leetcode.cn/leetbook/read/illustrate-lcof/xz0lse

A 剑指 Offer 21. 调整数组顺序使奇数位于偶数前面 - 解决方案

题目描述

输入一个整数数组，实现一个函数来调整该数组中数字的顺序，使得所有奇数位于数组的前半部分，所有偶数位于数组的后半部分。

示例：

输入：nums = [1, 2, 3, 4]

输出：[1, 3, 2, 4]

注：[3, 1, 2, 4] 也是正确的答案之一。

提示：

1. `1 <= nums.length <= 50000`
2. `1 <= nums[i] <= 10000`

解题方案

思路

- 标签：双指针
- 整体思路：首先指定前指针 `start` 和后指针 `end`，然后前指针定位偶数，后指针定位奇数，定位到之后将两个值互换，直到数组遍历完成
- 时间复杂度： $O(n)$ $O(n)$ ，空间复杂度： $O(1)$ $O(1)$

算法流程

1. 初始化前指针 `start = 0`，后指针 `end = nums.length - 1`
2. 当 `start < end` 时表示该数组还未遍历完成，则继续进行奇数和偶数的交换
3. 当 `nums[start]` 为奇数时，则 `start++`，直到找到不为奇数的下标为止
4. 当 `nums[end]` 为偶数时，则 `end--`，直到找到不为偶数的下标为止
5. 交换 `nums[start]` 和 `nums[end]`，继续下一轮交换
6. 返回 `nums`，即为交换后的结果

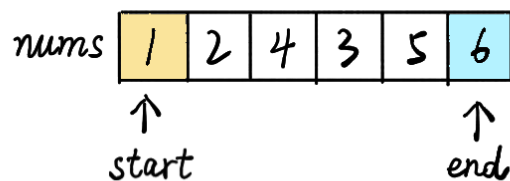
代码

```

class Solution {
    public int[] exchange(int[] nums) {
        int start = 0;
        int end = nums.length - 1;
        while(start < end) {
            while(start < end && (nums[start] % 2) == 1) {
                start++;
            }
            while(start < end && (nums[end] % 2) == 0) {
                end--;
            }
            int tmp = nums[start];
            nums[start] = nums[end];
            nums[end] = tmp;
        }
        return nums;
    }
}

```

画解



初始化 `start` 和 `end`

1 / 11

© 本 LeetBook 由「力扣」和作者共同制作和发行，版权所有侵权必究。本节内容是否对您有帮助？👍 12

讨论区

共 11 个讨论

最热 🔥



最好不过 💎 L1

来自陕西 2021-04-15



内部定义两个数组、进行拼接不就完了「代码块」

你的空间复杂度是 $O(n)$ ，准确来说是 $2n$ ，消耗两倍的空间



4



笨熊💎L5

来自广东2021-07-03



哈哈哈哈

刚说到 定义两个。。。 (面试官心想是指针)，结果一说数组，下一位



2



匿名用户

来自广东2021-04-19



内部定义两个数组、进行拼接不就完了「代码块」

面试官:“可以了，下一位”



1



MArtian

来自辽宁2021-08-12



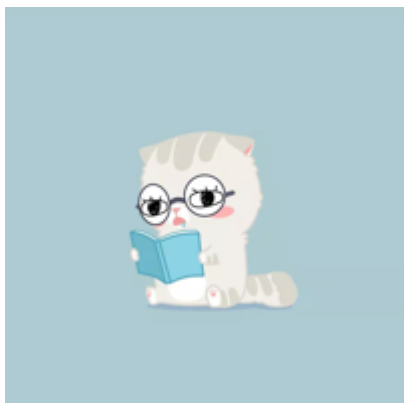
内部定义两个数组、进行拼接不就完了「代码块」

能用这些函数的话，算法题就没意义了



笨熊💎L5

来自广东2021-07-03



想知道为什么要“前指针定位偶数，后指针定位奇数，定位到之后将两个值互换”，能不能前面定位奇数、后面定位偶数，就省去互换的操作了呢？

前面是奇数是不用动的，不需要定位，只有前面是偶数，才可以，就是把前面的偶数找出来，把后面的奇数找出来，互换，这样前面都奇，后面都偶



努力成长的小白

来自上海2021-06-09

想知道为什么要“前指针定位偶数，后指针定位奇数，定位到之后将两个值互换”，能不能前面定位奇数、后面定位偶数，就省去互换的操作了呢？



tour1st



L1

来自河北2021-06-02

Go 时间击败44.12% 空间击败54.19%

```
func exchange(nums []int) []int {
    left, right := 0, len(nums)-1
    for left < right {
        for left < right && (nums[left]&1==1) {
            left++
        }
        for left < right && (nums[right]&1==0) {
            right--
        }
        nums[left], nums[right] = nums[right], nums[left]
    }
    return nums
}
```



太司马左边的高玩 L1

来自广东2021-06-01



内部定义两个数组、进行拼接不就完了「代码块」

这种题一般都是要求原地修改吧



匿名用户


来自荷兰2021-04-20



面试官:“可以了，下一位”

哈哈哈哈



点燃一支梦  L1

来自北京2021-04-08

内部定义两个数组、进行拼接不就完了

```
/**
 * @param {number[]} nums
 * @return {number[]}
 */
var exchange = function(nums) {
    let res1 = []
    let res2 = []

    for(let i = 0, len = nums.length; i < len; i++) {

        if(nums[i] % 2 !== 0) {
            res1.push( nums[i])
        }else {
            res2.push(nums[i])
        }
    }

    return res1.concat(res2)
};
```

