

## 第28题——调整数组顺序使奇数位于偶数前面

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：

0 <= nums.length <= 50000 1 <= nums[i] <= 10000

### 双指针法（1）

新建一个数组，大小相等，遍历nums，如果 %2 的结果为真，则表示该数字为奇数，放到新数组即可。

两个指针，一个指向数组头部，一个指向数组尾部

遍历之后返回新的数组即可

```
/**
 * Note: The returned array must be malloced, assume caller calls free().
 */
int* exchange(int* nums, int numsSize, int* returnSize){
    int *arr=calloc(numsSize,sizeof(int));
    int front=0;
    int rear=numsSize-1;
    for(int i=0;i<numsSize;i++)
    {
        if(nums[i]%2)
        {
            arr[front++]=nums[i];
        }
        else
        {
            arr[rear--]=nums[i];
        }
    }
    *returnSize=numsSize;
```

```
    return arr;
}
```

## 评论区大佬的双指针法(2)

思想:

1. 初始化:  $i, j$  双指针, 分别指向数组 `nums` 左右两端; 循环交换: 当  $i = j$  时跳出;
2. 指针  $i$  遇到奇数则执行  $i = i + 1$  跳过, 直到找到偶数; 指针  $j$  遇到偶数则执行  $j = j - 1$  跳过, 直到找到奇数; 交换 `nums[i]` 和 `nums[j]` 值;
3. 返回值: 返回已修改的 `nums` 数组。

```
class Solution {
public:
    vector<int> exchange(vector<int>& nums)
    {
        int i = 0, j = nums.size() - 1;
        while (i < j)
        {
            while(i < j && (nums[i] & 1) == 1) i++;
            while(i < j && (nums[j] & 1) == 0) j--;
            swap(nums[i], nums[j]);
        }
        return nums;
    }
};
```

作者: jyd

链接: <https://leetcode-cn.com/problems/diao-zheng-shu-zu-shun-xu-shi-qi-shu-wei-yu-ou-shu-qian-mian-lcof/solution/mian-shi-ti-21-diao-zheng-shu-zu-shun-xu-shi-qi-4/>

来源: 力扣 (LeetCode)

著作权归作者所有。商业转载请联系作者获得授权, 非商业转载请注明出处。