# 题目

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。

**示例 1:**

给定 nums = [3,2,2,3], val = 3,

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。

**示例 2:**

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。

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，不对实参作任何拷贝

int len = removeElement(nums, val);

// 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]);

}

# 分析

## 方法一：双指针

class Solution {

public:

int removeElement(vector<int>& nums, int val) {

int i = 0;

for(int j=0;j<nums.size();j++)

{

if(nums.at(j) != val)

{

nums[i] = nums[j];

i++;

}

}

return i;

}

};

说明：上述代码与LeetCode 26的不一样，26是删除重复的，因此第一个肯定不重复，只需要从第2个开始，这里是判断是否等于指定的val，因此需要从0开始遍历。