[27. 移除元素](https://leetcode-cn.com/problems/remove-element/)

难度简单533收藏分享切换为英文关注反馈

给你一个数组 *nums*和一个值 *val*，你需要 [**原地**](https://baike.baidu.com/item/%E5%8E%9F%E5%9C%B0%E7%AE%97%E6%B3%95) 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**](https://baike.baidu.com/item/%E5%8E%9F%E5%9C%B0%E7%AE%97%E6%B3%95)**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]);

}

我的通过：

public class Solution {

    public int RemoveElement(int[] nums, int val)

    {

        if(nums ==null || nums.Length == 0) return 0;

        int len = nums.Length;

        int i =0;

        int j =0;

        while(j < len)

        {

            if(nums[j] != val) nums[i++] = nums[j];

            j++;

        }

        return i ;

    }

}

我的不通过：

public class Solution {

    public int RemoveElement(int[] nums, int val)

    {

        if(nums ==null || nums.Length == 0) return 0;

        int len = nums.Length;

        int i =0;

        int j =0;

        while(j < len)

        {

            if(nums[j] != val) nums[i++] = nums[j++];

            //j++;

        }

        return i ;

    }

}

错误分析：

我第一遍写的时候没有把和val相等的情况考虑进来，不然j就不递增就会陷入死循环。后修改加入j++在循环体内。然后把nums[j++]改成nums[j]，通过。

参考代码：

#双指针法  
##public class Solution {  
public int RemoveElement(int[] nums, int val)  
{  
int currentIndex = 0;  
for (int i = 0; i < nums.Length; i++)  
{  
if (nums[i] != val)  
{  
nums[currentIndex] = nums[i];  
currentIndex++;  
}  
}  
return currentIndex;  
}

}