<https://leetcode-cn.com/explore/featured/card/top-interview-questions-easy/1/array/21/>

**删除排序数组中的重复项**

给定一个排序数组，你需要在[**原地**](http://baike.baidu.com/item/%E5%8E%9F%E5%9C%B0%E7%AE%97%E6%B3%95) 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**](https://baike.baidu.com/item/%E5%8E%9F%E5%9C%B0%E7%AE%97%E6%B3%95)**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]);

}

我的默写代码：

public class Solution {

public int RemoveDuplicates(int[] nums) {

//删除数组的重复值并且返回数组长度

//新数组长度记录

int i = 0 ;

//遍历标识

int j = 1;

for(;j<nums.Length;)

{

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

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

}

return i;

}

}

两个错误：

1、@@ if(nums[j] != nums[i]) j++; if(nums[j] == nums[i]) 逻辑错误 两者应该是相反的。如果I j 下标相同则继续遍历，不同则输入新的数组进行赋值。

2、@@ i不是输出的数组长度，应该是i+1

修改后：

public class Solution {

public int RemoveDuplicates(int[] nums) {

//删除数组的重复值并且返回数组长度

//新数组长度记录

int i = 0 ;

//遍历标识

int j = 1;

for(;j<nums.Length;)

{

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

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

}

return i+1;

}

}

结果：通过

参考代码：

public int RemoveDuplicates(int[] nums) {

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

int i = 0;

int j = 1;

int len = nums.Length;

while (j < len)

{

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

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

}

return i+1;

}

————————————————

版权声明：本文为CSDN博主「des\_null」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/moon\_goes/article/details/102763792