



链滴

从排序数组中删除重复项

作者: [someone31851](#)

原文链接: <https://ld246.com/article/1546234040666>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

给定一个排序数组，你需要在**原地**删除重复出现的元素，使得每个元素只出现一次，返回移除后组的新长度。

不要使用额外的数组空间，你必须在**原地修改输入数组**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]);
}
```

思路

使用双指针进行对比，如果 `num[i] == num[j]`，则指针不变，若不等于，则赋值 `num[i] = num[j]`，对比结束后，`i`的指针位置加 1 则为最终应该返回的值

```
class Solution {
    public int removeDuplicates(int[] nums) {
        if (nums == null || nums.length == 0) {
            return 0;
        }

        int length = 0;
        for (int i = 1; i < nums.length; i++) {
            if (nums[i] != nums[length]) {
                length++;
            }
        }
    }
}
```

```
        nums[length] = nums[i];  
        System.out.println(length + "==" + nums[length]);  
    }  
}  
return length + 1;  
}  
}
```