「删除有序数组重复项」的通解

# [一招秒杀所有同类型题] 非常容易理解的双指针

[Eason L5 发布于 2021-07-20775](https://leetcode-cn.com/u/eason734/)[[双指针](https://leetcode-cn.com/u/eason734/)](https://leetcode-cn.com/tag/two-pointers/)[[Python](https://leetcode-cn.com/u/eason734/)](https://leetcode-cn.com/topic/python/)[[Python3](https://leetcode-cn.com/u/eason734/)](https://leetcode-cn.com/topic/python3/)

大家好，我是Eason，今天我会教大家一套解决这类题目的双指针思维框架，烦请大家阅读过程中同时动手画画草图模拟出双指针移动的过程，我保证您事后定有收获！下面让我们开始分析题目:

# 解析

和这道题目的容易版本 (26.删除有序数组中的重复项) 不同的是，现在题目放宽了要求: 允许元素**重复两次**, 但是思路其实是一模一样的。因此在做本题之前，先让我们看（26.删除有序数组中的重复项）  
(请不要觉得简单，它们的思路真的完全一样, 因此让我们一点点递进)

### "第26题要求把数组中所有重复元素剔除，反过来想实质上就是要求把所有不重复元素保留"

注意! 上面一句话是此题的题眼，理解透彻了该题将迎刃而解

"把所有不重复元素进行保留" 可以通过双指针来实现 (该思维框架可以总结成模板运用到很多类似题)：即定义两个指针left和right，right从左到右把所有元素扫一遍，将不重复的元素赋给left的下一位

**详细步骤：**

1. left = 0，right = 0，right负责将所有元素从左到右扫一遍，left负责承接right找到的非重复元素
2. right不断右移，判断nums[right]是否等于nums[left]  
   (1) 如果相等, 说明right指向的元素是重复元素，不保留，right继续右移  
   (2) 如果不相等, 说明right指向的元素不是重复元素，保留，然后右移left一个单位，再把right的值赋给left，right继续右移
3. 重复以上两个步骤直到right扫完整个数组，最后返回left + 1即可

代码如下：

# 26. 删除有序数组中的重复项

class Solution:

def removeDuplicates(self, nums: List[int]) -> int:

left = 0

for right in range(len(nums)):

# 如果相等, 说明right指向的元素是重复元素，不保留

if nums[right] == nums[left]:

continue

# 如果不相等, 说明right指向的元素不是重复元素，保留，然后右移left一个单位，再把right的值赋给left

left += 1

nums[left] = nums[right]

return left + 1

再次强调，第26题的题眼为**题目要求把数组中所有重复元素剔除，反过来想实质上就是要求把所有不重复元素保留**  
一旦这个思路弄懂了，许多本题的一些同类型题也就不再困难，比如我们当前遇见的这道题: 80. 删除有序数组中的重复项 II  
和第26题不同，本题现在允许每个元素可以出现**至多两次**。然而思路是完全一样的，因此代码和第26题相比只有**两个**变动 (请看下面的代码注释)

# 80. 删除有序数组中的重复项 II

class Solution:

def removeDuplicates(self, nums: List[int]) -> int:

# 变动1: 由于元素可以重复2次，left现在从第二个元素开始，right从第三个元素开始

left = 1

for right in range(2, len(nums)):

# 变动2: 以前之和nums[left]比, 现在还要和nums[left - 1]比，（从而保证元素可以重复两次）说明right指向的元素是重复了三次元素，不保留

if nums[right] == nums[left] and nums[right] == nums[left - 1]:

continue

left += 1

nums[left] = nums[right]

return left + 1

大家现在应该可以发现，两份代码长的非常相似，所以我们可以进一步把改题目generlize到允许每一个元素出现**至多K次**

class Solution:

def removeDuplicates(self, nums: List[int], K: int) -> int:

left = K - 1

for right in range(K, len(nums)):

tag = True

for i in range(K):

tag \*= nums[right] == nums[left - i]

if tag:

continue

left += 1

nums[left] = nums[right]

return left + 1

如果把K设置为5，input设置为[1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,3,3,3,4,4,5], 得到的输出将为[1,1,1,1,1,2,2,2,2,2,3,3,3,4,4,5]，符合预期  
Graphical user interface, text, application, email

Description automatically generated

感谢阅读，如有帮助，烦请三连支持，谢谢！