4/8 Leetcode

题目：

给你一个 升序排列 的数组 nums ，请你 原地 删除重复出现的元素，使每个元素 只出现一次 ，返回删除后数组的新长度。元素的 相对顺序 应该保持 一致 。

由于在某些语言中不能改变数组的长度，所以必须将结果放在数组nums的第一部分。更规范地说，如果在删除重复项之后有 k 个元素，那么 nums 的前 k 个元素应该保存最终结果。

将最终结果插入 nums 的前 k 个位置后返回 k 。

不要使用额外的空间，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。

判题标准:

系统会用下面的代码来测试你的题解:

int[] nums = [...]; // 输入数组

int[] expectedNums = [...]; // 长度正确的期望答案

int k = removeDuplicates(nums); // 调用

assert k == expectedNums.length;

for (int i = 0; i < k; i++) {

assert nums[i] == expectedNums[i];

}

如果所有断言都通过，那么您的题解将被 通过。

示例 1：

输入：nums = [1,1,2]

输出：2, nums = [1,2,\_]

解释：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 。不需要考虑数组中超出新长度后面的元素。

示例 2：

输入：nums = [0,0,1,1,1,2,2,3,3,4]

输出：5, nums = [0,1,2,3,4]

解释：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 。不需要考虑数组中超出新长度后面的元素。

提示：

0 <= nums.length <= 3 \* 104

-104 <= nums[i] <= 104

nums 已按 升序 排列

来源：力扣（LeetCode）

链接：<https://leetcode-cn.com/problems/remove-duplicates-from-sorted-array>

思路：

一开始想用先遍历数组，找到不一样的数字然后存入新的数组，结果读题到后面发现，要在原有的数组上进行修改，我一下子没有了思路，思考了很久也不知道要怎么做，所以还是查看了题解。

现在知道了，要在原有的数组上进行修改，则需要用到双指针，一个用于遍历数组，另一个用于对原本的数组进行修改，由于定义函数用了引用，所以甚至都不需要返回任何值。看了一下题解也稍微有了思路，自己在稍微摸索一下，就通过题目了。

总结：

我学到了新的算法：双指针。

这是我第一次学到这个算法，知道了原来要对原本的数组进行修改，只需定义一个双指针，便可以轻松的完成，又学习到了一个新的算法，之后遇到类似的题目我也可以更快的有了思路，更快的解题了，算是有所收获！