# 26. Remove Duplicates from Sorted Array

Given a sorted array nums, remove the duplicates in-place such that each element appear only once and return the new length.

Do not allocate extra space for another array, you must do this by modifying the input array in-place **with O(1) extra memory**.

*Example 1:*

Given nums = [1,1,2],

Your function should return length = 2, with the first two elements of nums being 1 and 2 respectively.

It doesn't matter what you leave beyond the returned length.

*Example 2*:

Given nums = [0,0,1,1,1,2,2,3,3,4],

Your function should return length = 5, with the first five elements of nums being modified to 0, 1, 2, 3, and 4 respectively.

It doesn't matter what values are set beyond the returned length.

**Clarification**:

Confused why the returned value is an integer but your answer is an array?

Note that the input array is passed in by reference, which means modification to the input array will be known to the caller as well.

Internally you can think of this:

// nums is passed in by reference. (i.e., without making a copy)

int len = removeDuplicates(nums);

// any modification to nums in your function would be known by the caller.

// using the length returned by your function, it prints the first len elements.

**for (int i = 0; i < len; i++) {**

**print(nums[i]);**

**}**

## 算法1：优秀。

思路：很好理解。两个索引变量，i用于遍历整个数组，j用于标记非重复元素的索引，同时也代表着非重复元素个数。

class Solution {

public int removeDuplicates(int[] nums) {

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

int j = 1;

**for(int i = 1;i < nums.length;i++){**

**if(nums[i] != nums[i-1])**

**nums[j++] = nums[i];**

**}**

return j;

}

}

# 27. Remove Element

Given an array nums and a value val, remove all instances of that value in-place and return the new length.

Do not allocate extra space for another array, you must do this by modifying the input array in-place with O(1) extra memory.

The order of elements can be changed. It doesn't matter what you leave beyond the new length.

Example 1:

Given nums = [3,2,2,3], val = 3,

Your function should return length = 2, with the first two elements of nums being 2. **[2,2]**

It doesn't matter what you leave beyond the returned length.

Example 2:

Given nums = [0,1,2,2,3,0,4,2], val = 2,

Your function should return length = 5, with the first five elements of nums containing 0, 1, 3, 0, and 4. **[0,1,3,0,4]**

Note that the order of those five elements can be arbitrary.

It doesn't matter what values are set beyond the returned length.

**Clarification**:

Confused why the returned value is an integer but your answer is an array?

Note that the input array is passed in by reference, which means modification to the input array will be known to the caller as well.

Internally you can think of this:

// nums is passed in by reference. (i.e., without making a copy)

int len = removeElement(nums, val);

// any modification to nums in your function would be known by the caller.

// using the length returned by your function, it prints the first len elements.

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

print(nums[i]);

}

## 算法1

思路：两个索引i和j，索引i用于遍历整个数组，当前元素与val不相等，就拷贝到j指定位置，j不断递增。

class Solution {

public int removeElement(int[] nums, int val) {

int j = 0;

**for(int i = 0;i<nums.length;i++){**

**if(nums[i] != val){**

**nums[j++] =nums[i];**

**}**

**}**

return j;

}

}