

26. Remove Duplicates from Sorted Array

Description

Hints

Submissions

Discuss

Solution

Pick One

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:

```
public class L26 {
    public int removeDuplicates(int[] nums) {
        int length = nums.length;
        if(nums == null || length < 1)
            return length;
        int count = 1;
        for(int i = 1; i < length; i++) {
            if(nums[i] != nums[i - 1])
                nums[count++] = nums[i];
        }
        return count;
    }
}
```