Remove Duplicates from Sorted Array

# Question

Given a sorted array nums, remove the duplicates in-place such that each element appears only once and returns 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.

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 a modification to the input array will be known to the caller as well.

**Example 1:**

Input: nums = [1,1,2]

Output: 2, nums = [1,2]

Explanation: 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:**

Input: nums = [0,0,1,1,1,2,2,3,3,4]

Output: 5, nums = [0,1,2,3,4]

Explanation: 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.

# Pseudo Code

Declare and Initialize appropriate Variables

If size of the array is 0 or one

Return the size of the array

Run the First(Outer) For Loop

If the duplicate is encountered

Run the Second(Inner) For Loop

Shift all the Elements from that position to left

Decrement the Value of i

Decrement the Value of length

Return length

# Source Code

## V 1.0

1. int removeDuplicates(int\* nums, int numsSize){
3. int length = (numsSize - 1);
5. if(numsSize == 0 || numsSize == 1) {
6. return numsSize;
7. }
9. for(int i=0 ; i<length ; i++) {
11. if(nums[i] == nums[i + 1]) {
13. for(int j=(i+1) ; j<length ; j++) {
15. nums[j] = nums[j + 1];
16. }
17. i--;
18. length--;
19. }
20. }
21. return (length + 1);
22. }