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.

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]);

}

**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 variables pointerOne and pointerTwo to 0

Run the for loop from 0 to the size of the Array

If Adjacent Elements are Equal

Increment pointerTwo variable by one

Else

nums[pointerOne + 1] = nums[pointerTwo + 1];

Return (pointerOne + 1)

# Source Code

## v 1.0

1. int removeDuplicates(int\* nums, int numsSize){
3. if(numsSize < 2) {
4. return numsSize;
5. }
7. int pointerOne = 0, pointerTwo = 0;
8. for(int i=0 ; i<(numsSize - 1) ; i++) {
10. if(nums[i] == nums[i + 1]) {
11. pointerTwo++;
12. }
13. else {
14. nums[++pointerOne] = nums[++pointerTwo];
15. }
16. }
17. return (++pointerOne);
18. }