1. Remove Duplicates from Sorted Array II

Medium

Given a sorted array *nums*, remove the duplicates [**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) such that duplicates appeared at most *twice* 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 = [1,1,1,2,2,3],  
  
Your function should return length = 5, with the first five elements of nums being 1, 1, 2, 2 and 3 respectively.  
  
It doesn't matter what you leave beyond the returned length.

**Example 2:**

Given nums = [0,0,1,1,1,1,2,3,3],  
  
Your function should return length = 7, with the first seven elements of nums being modified to 0, 0, 1, 1, 2, 3 and 3 respectively.  
  
It doesn't matter what values are set beyond the returned length.

**Solution**

class Solution {  
public:  
 int removeDuplicates(vector<int>& nums) {  
 int len = nums.size();  
 int i = 0, j = 0;  
 while(j < len){  
 if(i < 2 || nums[j] != nums[i-1] || nums[j] != nums[i-2]){  
 nums[i] = nums[j];  
 i++;  
 }  
 j++;  
 }  
 return i;  
 }  
};