26. Remove Duplicates from Sorted Array

Given an integer array nums sorted in non-decreasing order, remove the duplicates in-place such that each unique e

Consider the number of unique elements of nums to be k, to get accepted, you need to do the following things:

Change the array nums such that the first k elements of nums contain the unique elements in the order they were pr
Return k.
Custom Judge:

The judge will test your solution with the following code:

```
int[] nums = [...]; // Input array
int[] expectedNums = [...]; // The expected answer with correct length

int k = removeDuplicates(nums); // Calls your implementation

assert k == expectedNums.length;
for (int i = 0; i < k; i++) {
    assert nums[i] == expectedNums[i];
}
```

If all assertions pass, then your solution will be accepted.

Example 1:

```
Input: nums = [1,1,2]
Output: 2, nums = [1,2,_]
Explanation: Your function should return k = 2, with the first two elements of nums being 1 and 2 respectively.
It does not matter what you leave beyond the returned k (hence they are underscores).
```

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 k = 5, with the first five elements of nums being 0, 1, 2, 3, and 4 respe
It does not matter what you leave beyond the returned k (hence they are underscores).
```

Constraints:

```
1 <= nums.length <= 3 * 104
-100 <= nums[i] <= 100
nums is sorted in non-decreasing order.
```

Solutions

Brute-Force

```
var removeDuplicates = function(nums) {
    if (nums.length === 0) return 0;
```

```
    let k = 0; // Pointer for the position of the next unique element

    for (let i = 0; i < nums.length; i++) {
        let isDuplicate = false;
        // Check if nums[i] is already in the unique part of the array
        for (let j = 0; j < k; j++) {
            if (nums[i] === nums[j]) {
                isDuplicate = true;
                break;
            }
        }
        // If it's not a duplicate, place it in the unique part of the array
        if (!isDuplicate) {
            nums[k] = nums[i];
            k++;
        }
    }

    return k;
};
```

Optimal - Using Two Pointers

```
var removeDuplicates = function(nums) {
    let i=0;j=1;

    while(j<nums.length){
        if(nums[i]!==nums[j]){
            i++
            nums[i]=nums[j]
        }
        j++
    }
    return i+1
};
```