

# Array #7

## Leetcode #26

### Remove Duplicates from Sorted Array

<https://leetcode.com/problems/remove-duplicates-from-sorted-array/>

Given an integer array nums sorted in non-decreasing order, remove the duplicates in-place such that each unique element appears only once. The relative ordering of the elements should be kept the same. Then return the number of unique elements in nums.

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 present in nums initially. The remaining elements of nums are not important as well as the size of nums
- return k

Example 1:

Input: nums = [1, 1, 2]

Output: 2, nums = [1, 2, \_]

Example 2:

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

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

Constraints:

$0 \leq \text{nums.length} \leq 3 \times 10^4$

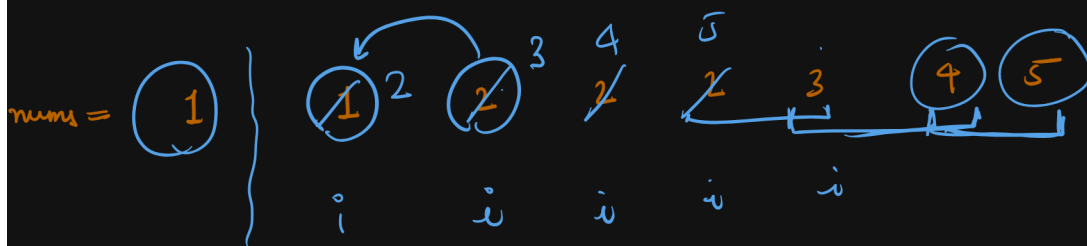
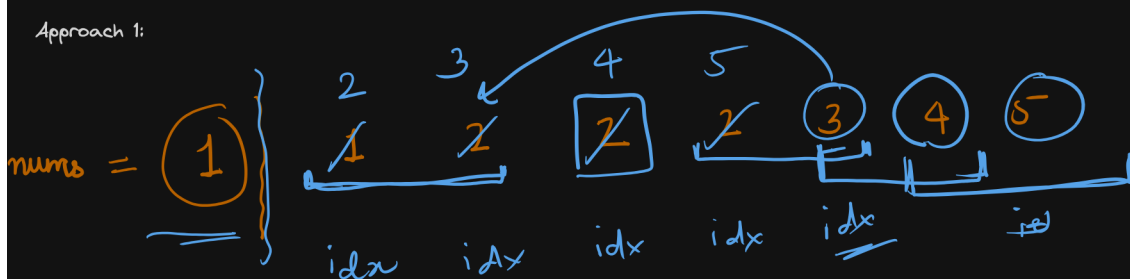
$-100 \leq \text{nums}[i] \leq 100$

nums is sorted in non-decreasing order

Companies:

Amazon, Meta, Google, Microsoft, Bloomberg, etc.

Approach 1:

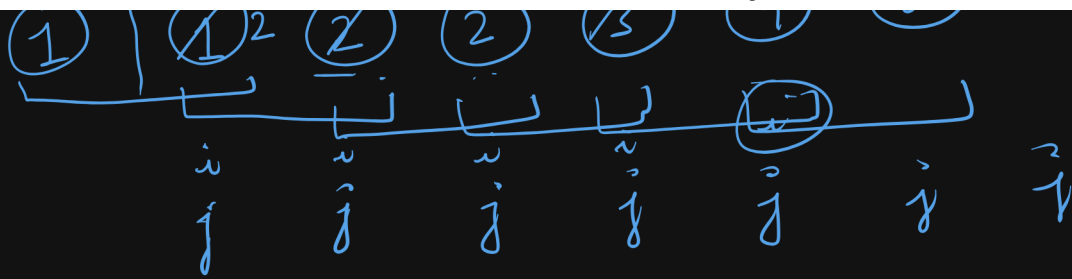


Time complexity:

$O(N)$

Space Complexity:

$O(1)$



```

class Solution {
    public int removeDuplicates(int[] nums) {
        int i = 1;
        // 1 2 3 3 3 4
        for (int j = 1; j < nums.length; j++) {
            if (nums[j - 1] != nums[j]) {
                nums[i] = nums[j];
                i++;
            }
        }

        return i;
    }
}

```