

Leetcode Problem 1. (Easy)

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 order** 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` 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`.

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 respectively.

It does not matter what you leave beyond the returned `k` (hence they are underscores).

Constraints:

- $1 \leq \text{nums.length} \leq 3 \times 10^4$

- $-100 \leq \text{nums}[i] \leq 100$
- `nums` is sorted in **non-decreasing** order.

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

```
class Solution {
    public int removeDuplicates(int[] nums)
    {

        if (nums.length == 0)
        {
            return 0;
        }
        int k = 1;
        for (int i = 1; i < nums.length; i++)
        {
            if (nums[i] != nums[k - 1])
            {
                nums[k++] = nums[i];
            }
        }
        return k;
    }
}
```

The screenshot shows the LeetCode interface for a submission. On the left, the 'Submissions' tab is active, showing a green 'Accepted' status and a 'Next question' button. Below this, there are links to other problems: '27. Remove Element', '80. Remove Duplicates from Sorted Array II', and '2460. Apply Operations to an Array'. The main content area on the right displays the user's profile 'Sakib Rahman' with a 'Details' button and a '+ Solution' button. Below the profile, there is a performance chart showing 'Runtime 1 ms', 'Beats 99.97%', 'Memory 44.3 MB', and 'Beats 26.26%'. A 'Notes' section with a text input field is also present. At the bottom, there is a 'Related Tags' section with a 'Select tags' dropdown. The code editor at the bottom right shows the same Java code as in the previous block. At the very bottom, there are buttons for 'Console', 'Run', and 'Submit'.