

LEET CODE - 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` 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`.

CODE:

```
int removeDuplicates(int* nums, int numsSize) {
    if (numsSize == 0) return 0;
    int x = 0;
    for (int y = 1; y < numsSize; y++) {
        if (nums[y] != nums[x]) {
            x++;
            nums[x] = nums[y];
        }
    }
    return x + 1;
}
```

OUTPUT :

Accepted Runtime: 0 ms

- Case 1
- Case 2

Input

```
nums =  
[1,1,2]
```

Output

```
[1,2]
```

Expected

```
[1,2]
```

Accepted Runtime: 0 ms

- Case 1
- Case 2

Input

```
nums =  
[0,0,1,1,1,2,2,3,3,4]
```

Output

```
[0,1,2,3,4]
```

Expected

```
[0,1,2,3,4]
```