**LeetCode:**

**26. Remove duplicates from sorted array**

<https://leetcode.com/problems/remove-duplicates-from-sorted-array/description/?envType=problem-list-v2&envId=array&difficulty=EASY>

**My solution:**

class Solution:

def removeDuplicates(self, nums: List[int]) -> int:

if not nums:

return 0

i = 0

for j in range(i+1, len(nums)):

if nums[i] != nums[j]:

i +=1

nums[i] = nums[j]

return i+1

**Explanation:**

**Example:**

[1,1,2,3,3]

When i is 0, j is set to 1. Since nums[i] is equal to nums[j], j is incremented to 2, and i is incremented by one, making i equal to 1. We then swap nums[i] with nums[j] to move unique elements to the front of the array. This process continues, ensuring that unique elements are placed at the beginning of the array. Finally, we return i + 1 because i was initialized to 0. Adding 1 to i gives us the length of the array containing only unique elements, as it traverses to the index of the last unique element and then adds one to include the count of all unique elements.

**27. Remove Elements**

<https://leetcode.com/problems/remove-element/description/?envType=problem-list-v2&envId=array&difficulty=EASY>

class Solution:

def removeElement(self, nums: List[int], val: int) -> int:

if not nums:

return 0

count = 0

for i in range(0, len(nums)):

if nums[i] != val:

nums[count] = nums[i]

count+=1

return count

**Explanation:**

Similar to the above remove duplicates from the sorted array problem, here we are asked to return the count