**27. Remove Element (LeetCode)**

Given an array and a value, remove all instances of that value [**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) and return the new length.

Do not allocate extra space for another array, you must do this by **modifying the input array**[**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) with O(1) extra memory.

The order of elements can be changed. It doesn't matter what you leave beyond the new length.

**Example:**

Given **nums** = [3,2,2,3], **val** = 3,

Your function should return length = 2, with the first two elements of *nums* being 2.

🡨---------------------------------------------------------------------------------------------------------------🡪

*class Solution {*

*public int removeElement(int[] nums, int val) {*

*int length = nums.length;*

*int valInstances = 0;*

*for (int i = 0; i < length; i++){*

*if(nums[i] == val){*

*valInstances++;*

*nums[i]= nums[length-1];*

*length--;*

*i--;*

*}*

*}*

*return (nums.length - valInstances);*

*}*

*}*

🡨------------------------------------------------------------------------------------------------------🡪