Array #6

Leetcode #27

Remove Element

https://leetcode.com/problems/remove-element/description/

Given an integer array nums and an integer val, remove all occurences of val in nums in-place. The order of the elements may be changed. Then return the number of elements in nums which are not equal to val

Consider the number of elements in nums which are not equal to val be k, to get accepted, you need to do the following things:
- Change the array nums such that the first k elements of nums contains the elements which are not equal to val.
The remaining elements of nums are not important as well as the size of nums
- return k

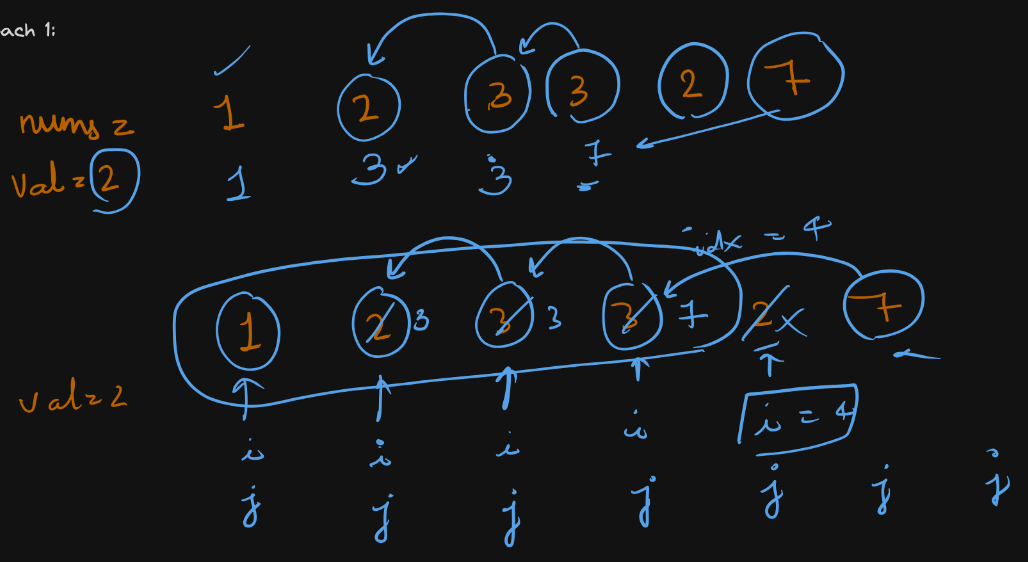
Example 1:
Input: nums = [3, 2, 2, 3], val = 3
Output: 2, nums = [2, 2, _, _]

Example 2:
Input: nums = [0, 1, 2, 2, 3, 0, 4, 2], val = 2
Output: 5, nums = [0, 1, 4, 0, 3, _, _, _]

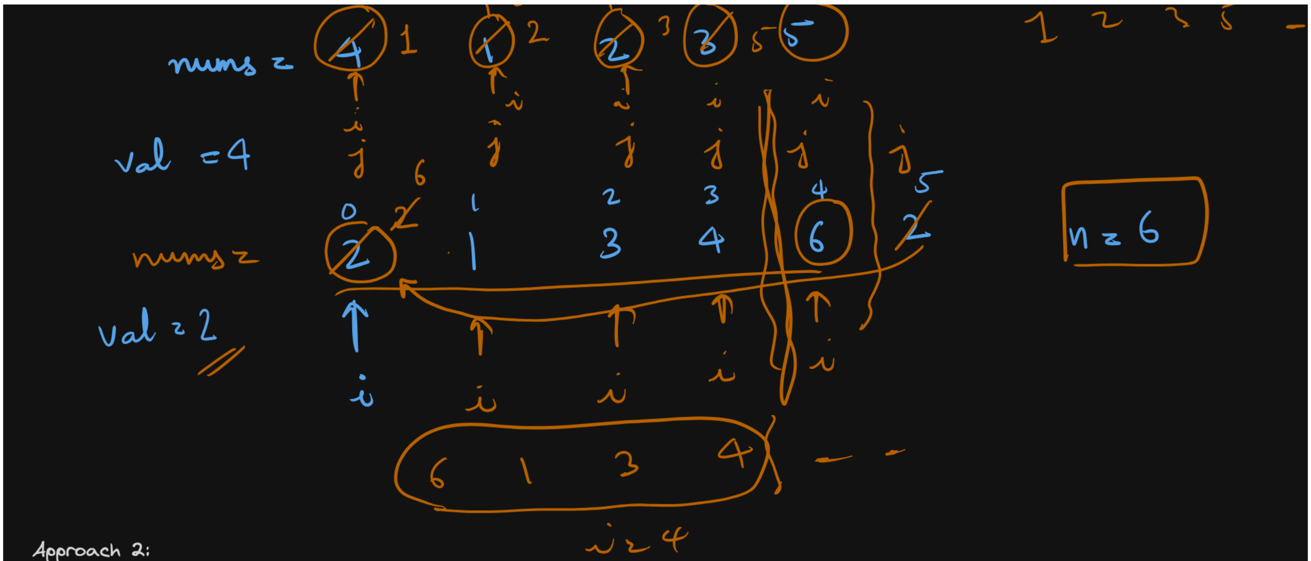
Constraints:
0 <= nums.length <= 100
0 <= nums[i] <= 50
0 <= val <= 100

Companies:
Amazon, Google, Adobe, Bloomberg, Meta, Microsoft, Apple, etc.

Approach 1:



Time complexity:
O(N)
Space Complexity:
O(1)



Approach 2:

Time complexity:
O(n) ✓
Space Complexity:
O(1) ✓

```
class Solution {
    public int removeElement(int[] nums, int val) {
        // approach 1
        int i = 0;

        for (int j = 0; j < nums.length; j++) {
            if (nums[j] != val) {
                nums[i] = nums[j];
                i++;
            }
        }

        return i;
    }
}


class Solution {
    public int removeElement(int[] nums, int val) {
        // approach 2
        int n = nums.length;

        int i = 0;

        while (i < n) {
            if (nums[i] == val) {
                nums[i] = nums[n - 1];
                n--;
            } else {
                i++;
            }
        }
```

```
        return n;
    }
}
```