/*Given an array and a value, remove all instances of that value in-place and return the new length.

Do not allocate extra space for another array, you must do this by modifying the input array in-place 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.*/

-
- 思想：
- (1) 双指针思想：双指针思想，一个头指针指向数组的头，一个尾指针指向数组的尾；当头指针指向val值的时候，移动尾指针知道它指向非val值，将非val值转移到头指针处
- (2) 巧妙之处是
- 1、当头指针未指向val值时，不需要进行任何操作；
- 2、只需将尾指针的值转移到头指针处，不需要把头指针的值转到尾指针处，因为题目没有要求“垃圾”部分的值
- (3)这道题和26题的区别是：26题要求是有序的，所以只能用两个同向指针遍历数组；而本题目无序，所以可以使用相遇指针加快速度

```
public int removeElement(int[] nums, int val) {
    int left = 0;
    int right = nums.length - 1;
    int len = nums.length;
    while (left <= right) {
        if (nums[left] == val) {
            while (left < right && nums[right] == val) {
                len--;
                right--;
            }
            nums[left] = nums[right];
            right--;
            len--;
        }
        left++;
    }
    return len;
}
```