27 移除元素

问题描述：原地移除数组中所有等于val的值，返回数组的新大小

注意：数组元素在内存地址中是连续的，不能删除，只能覆盖

暴力解法

两层for循环，外层用于找到值等于val的元素，内层用于更新（覆盖）数组

```
class Solution {
public:
    int removeElement(vector<int>& nums, int val) {
        int sz = nums.size();
        for(int i = 0;i<sz;++i){
            if(nums[i]==val){
                for(int j = i+1;j<sz;j++){
                    nums[j-1] = nums[j];
                }
                --i; // 下标i之后的元素都向前移了一位，所以i-1
                --sz;
            }
        }
        return sz;
    }
};
```

时间复杂度O(n2)

空间复杂度O(1)

双指针法(快慢指针法)

定义一个快指针和一个慢指针，在一个for循环下完成

```
    class Solution {
public:
    int removeElement(vector<int>& nums, int val) {
        int slowIndex = 0;
        // 快指针用来遍历整个数组
        for(int fastIndex = 0;fastIndex <nums.size();++fastIndex){
            // 慢指针用来接收值不等于val的元素
            if(nums[fastIndex] != val){
```

相关题目：26,283,844,977

---

26. 原地删除有序数组中的重复项

```
class Solution {
public:
    int removeDuplicates(vector<int>& nums) {
        int slowIndex = 0;
        for(int fastIndex =0;fastIndex!= nums.size();++fastIndex){

            // 如果快慢指针对应的值不相等，才把快指针的值赋给慢指针下一位的元素
            if(nums[slowIndex] != nums[fastIndex]){
                nums[++slowIndex] = nums[fastIndex];
            }
        }
        return ++slowIndex;
    }
};
```

---

283 移动0

---

844 比较含空格的字符串

```
class Solution {
public:
    bool backspaceCompare(string s, string t) {
        s = string_updata(s);
        t = string_updata(t);
        return s==t;
    }

private:
    string string_updata(string s){
        int slowIndex = 0;
        int sz = s.size();
        string str;
        for(int fastIndex = 0;fastIndex != s.size(); ++fastIndex){
            if(s[fastIndex] != '#'){ // 没有遇到#的情况
                s[slowIndex++] = s[fastIndex];
            } else{ // 遇到#的情况
                if(slowIndex!=0){ // 根据当前慢指针是否为0下标做进一步的细分
                    --slowIndex;
                    sz-=2; // 如果#之前有元素大小减2
                } else{
                    sz-=1; // 如果#之前没有元素大小减1
                }

            }
        }
        // if(sz==0) return ""; sz==0,str没有push_back,str原本就是空字符串，不需要这一
步
        for(int i =0;i<sz;++i){
            str.push_back(s[i]);
        }
        return str;
    }
};
```

977有序数组的平方

```
class Solution {
public:
    vector<int> sortedSquares(vector<int>& nums) {
        int n = nums.size();

        vector<int> ans(n); // 指定大小是为了用下标访问
```

In [ ]: