题目：151.翻转字符串里的单词

给定一个字符串，逐个翻转字符串中的每个单词。

示例 1：

输入："the sky is blue"

输出："blue is sky the"

示例 2：

输入：" hello world! "

输出："world! hello"

解释：输入字符串可以在前面或者后面包含多余的空格，但是反转后的字符不能包括。

> 如果是**python**那么可以对字符串进行split分割，然后创建新的String,最后把单词倒叙输出

「不要使用辅助空间，空间复杂度要求为O(1)。」即只能在源字符串上下功夫

(2)非常美妙的花式

> 我们可以先对**整个字符串进行翻转**，此时单词之间顺序发生反转，同时单词内部的字符也发生反转，在**对每个单词内部的字符进行反转**


```
对于移除空格的方法可以有
void removeExtrasSpaces(string& s){
```

```
for(int i=1;i<s.size();i++){
if(s[i]==s[i-1]&&s[i]==' '){
s.erase(s.begin()+i);
}
}
int i=0;
while(s[i]==' ') s.erase(s.begin()+i);
 int j=1;
 while(s[s.size()-j]==' '){
 s.erase(s.end()-j);
 }
}
以上的方法时间复杂度为O(n^2)因为数组的删除时间复杂度为O(n)外面再加一个循环此时复杂度为
O(n^2);
```

可以用双指针(快慢指针)实现数组移除元素：

编号：27. 移除元素

给你一个数组 nums 和一个值 val，你需要 原地 移除所有数值等于 val 的元素，并返回移除后数组的新长度。

不要使用额外的数组空间，你必须仅使用 O(1) 额外空间并「原地」修改输入数组。

元素的顺序可以改变。你不需要考虑数组中超出新长度后面的元素。

示例 1:
给定 nums = [3,2,2,3], val = 3,
函数应该返回新的长度 2, 并且 nums 中的前两个元素均为 2。
你不需要考虑数组中超出新长度后面的元素。

示例 2:
给定 nums = [0,1,2,2,3,0,4,2], val = 2,
函数应该返回新的长度 5, 并且 nums 中的前五个元素为 0, 1, 3, 0, 4。

> 多余的元素删除不就得了，要知道**数组在内存地址中是连续得**,不能**单独删除数组中某个元素，只能覆盖**
> **way1:用暴力解法但是非常慢**
> **way2:双指针法通过一个快指针和慢指针在一个for循环下完成两个for循环**

```
暴力:
class Solution {
```

```
public:
    int removeElement(vector<int>& nums, int val) {
            int size=nums.size();
            for(int i=0;i<size;i++){
                if(nums[i]==val){
                    for(int j=i;j<size-1;j++){
                        nums[j]=nums[j+1];
                    }
                    size--;
                    i--;
                }
            }
            return size;
    }
};

快慢指针:
 slowIndex:永远指向新数组的最后一个字符的后一个字符
 class Solution {
public:
    int removeElement(vector<int>& nums, int val) {
 int slowIndex=0;
 for(int fastIndex=0;fastIndex<nums.size();fastIndex++){
 if(val!=nums[fastIndex]){
 nums[slowIndex++]=nums[fastIndex];
 }
 }
 return slowIndex;
    }
};
```

现在跳转回去151题目

```
//移除冗余空格:使用双指针(快慢指针法)O(n)的算法
void removeExtraSpace(string& s){
 int slowIndex=0,fastIndex=0;//定义快慢指针,slowIndex的大小为新字符串的实际大小
 //去掉字符串前面的空格
 while(s.size()>0&&fastIndex<s.size()&&s[fastIndex]==' ') fastIndex++;
 //去除字符串中间的空格
 for(;fastIndex<s.size();fastIndex++){
```

```
 if(fastIndex-1>0&&
 s[fastIndex]==s[fastIndex-1]
  &&s[fastIndex]==' ') continue;
  else{
  s[slowIndex++]=s[fastIndex];
  }
  }
  if(slowIndex-1>0&&s[slowIndex-1]==' '){
  s.resize(slowIndex-1);
  }else{
  s.resize(slowIndex);
  }
 }
 //定义一个反转字符串s中左闭右闭的区间[start,end]
 void reverse(string& s,int start,int end){
  while(start<end)swap(s[start++],s[end--]);
 }
 string reverseWords(string s) {
  removeExtraSpace(s);//去掉冗余的空格
  reverse(s,0,s.size()-1);
  int fastIndex=0,slowIndex=0;
  for(;fastIndex<s.size();fastIndex++){
  if(s[fastIndex]==' '){
  //反转[slowIndex,fastIndex-1]组成的单词
  reverse(s,slowIndex,fastIndex-1);
  slowIndex=fastIndex+1;
  }
  }
  reverse(s,slowIndex,fastIndex-1);
  return s;
 }
```

局部反转+整体反转会有意想不到的效果！

题目：剑指Offer58-II.左旋转字符串

字符串的左旋转操作是把字符串前面的若干个字符转移到字符串的尾部。请定义一个函数实现字符串左旋转操作的功能。比如，输入字符串"abcdefg"和数字2，该函数将返回左旋转两位得到的结果"cdefgab"。

示例 1：
输入: s = "abcdefg", k = 2
输出: "cdefgab"

示例 2：
输入: s = "lrloseumgh", k = 6
输出: "umghlrlose"
限制：
1 <= k < s.length <= 10000

> **「不能申请额外空间，只能在本串上操作」**
> 对于字符串反转的问题我们可以利用全局反转+局部反转达到效果，反转的方法则用双指针方法
> 此时我们要选择是先局部反转还是全局反转比较好
> **此时选择局部反转因为局部空间是题目给出如果先全局反转则还算局部空间**

```
class Solution {
public:
    string reverseLeftWords(string s, int n) {
        //反转前n个字符
        int left=0,right=n-1;
        while(left<right){
            swap(s[left++],s[right--]);
        }
        //反转下标为n的字符到字符串s
         left=n;
         right=s.size()-1;
         while(left<right){
             swap(s[left++],s[right--]);
         }
         //在对s进行全局反转
         left=0;
         right=s.size()-1;
         while(left<right){
             swap(s[left++],s[right--]);
```

```
20          }
21          return s;
22      }
23  };
```

总结: