

(1) 这道题目，使用库函数一行代码搞定

题目：344. 反转字符串

编写一个函数，其作用是将输入的字符串反转过来。输入字符串以字符数组 `char[]` 的形式给出。

不要给另外的数组分配额外的空间，你必须原地修改输入数组、使用 $O(1)$ 的额外空间解决这一问题。

你可以假设数组中的所有字符都是 ASCII 码表中的可打印字符。

示例 1：

输入：["h","e","l","l","o"] 输出：["o","l","l","e","h"] 示例 2：

输入：["H","a","n","n","a","h"] 输出：["h","a","n","n","a","H"]

对于字符串，我们定义两个指针，一个指针在字符串开始，另一个指针在字符串的最后实现交换

```
1 class Solution {
2 public:
3     void reverseString(vector<char>& s) {
4         int left=0;
5         int right=s.size()-1;
6         while(left<right){
7             char tmp=s[left];
8             s[left]=s[right];
9             s[right]=tmp;
10            left++;
11            right--;
12        }
13    }
14 };
```

(2)我要花式反转

题目：541. 反转字符串II

给定一个字符串 s 和一个整数 k ，你需要对从字符串开头算起的每隔 $2k$ 个字符的前 k 个字符进行反转。

如果剩余字符少于 k 个，则将剩余字符全部反转。

如果剩余字符小于 $2k$ 但大于或等于 k 个，则反转前 k 个字符，其余字符保持原样。

示例：

输入： $s = \text{"abcdefg"}$, $k = 2$

输出： "bacdfeg"

该题目本身是一个模拟题目

```
1 class Solution {
2 public:
3     string reverseStr(string s, int k) {
4         int left=0,right=2*k-1,mid=left+(right-left)/2;
5         //注意:前k个字符区间为[left,mid],2k个字符的区间为[left,right]
6         while(right<s.size()){
7             int l=left,r=mid;
8             while(l<r){
9                 char tmp=s[l];
10                s[l]=s[r];
11                s[r]=tmp;
12                l++;
13                r--;
14            }
15            left+=2*k;
16            mid+=2*k;
17            right+=2*k;
18        }
19        right=min(mid,s.size()-1);
20        while(left<right){
21            char c=s[left];
22            s[left]=s[right];
23            s[right]=c;
24            left++;
25            right--;
26        }
```

```
27 return s;  
28 }  
29 };
```

执行结果: **通过** [显示详情](#) >

执行用时: **8 ms** , 在所有 C++ 提交中击败了 **20.32%** 的用户

内存消耗: **7.2 MB** , 在所有 C++ 提交中击败了 **30.61%** 的用户

炫耀一下:



(3)遇到对字符串或者数组做填充或删除的操作时，都要想想从后向前操作怎么样

题目：剑指Offer 05.替换空格

请实现一个函数，把字符串 `s` 中的每个空格替换成"%20"。

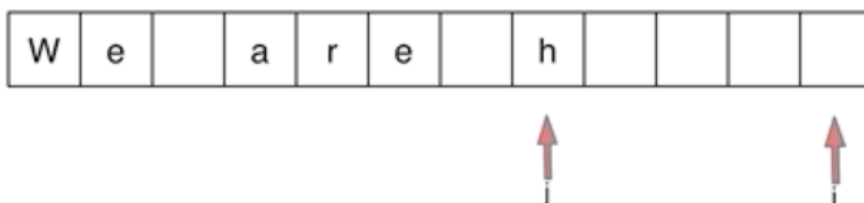
示例 1：

输入：s = "We are happy."

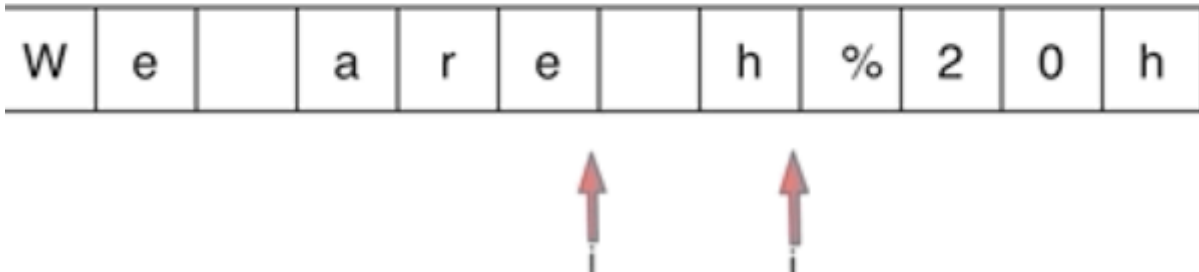
输出："We%20are%20happy."

思路：首先扩充数组到每个空格替换成"%20"之后的大小,然后从后向前替换空格，也就是双指针法,i指向新长度的末尾，j指向旧长度的末尾

输入：s = "We are h"



输入: s = "We are h"



```
1 class Solution {
2 public:
3     string replaceSpace(string s) {
4         int count=0,oldSize=s.size(),newSize=0;
5         for(int i=0;i<s.size();i++){
6             if(s[i]==' ') count++;
7         }
8         //调整字符串的大小
9         s.resize(s.size()+count*2);
10        newSize=s.size();
11        for(int i=newSize-1,j=oldSize-1;j<i;j--,i--){
12            if(s[j]!=' '){
13                s[i]=s[j];
14            }else{
15                s[i]='0';
16                s[i-1]='2';
17                s[i-2]='%';
18                i-=2;
19            }
20        }
21        return s;
22    }
23 };
```