1. Reverse Words in a String

Given an input string, reverse the string word by word.

**Example 1:**

Input: "the sky is blue"  
Output: "blue is sky the"

**Example 2:**

Input: " hello world! "  
Output: "world! hello"  
Explanation: Your reversed string should not contain leading or trailing spaces.

**Example 3:**

Input: "a good example"  
Output: "example good a"  
Explanation: You need to reduce multiple spaces between two words to a single space in the reversed string.

**Note:**

* A word is defined as a sequence of non-space characters.
* Input string may contain leading or trailing spaces. However, your reversed string should not contain leading or trailing spaces.
* You need to reduce multiple spaces between two words to a single space in the reversed string.

**解法1** 分词，然后用字符串拼接

class Solution {  
public:  
 string reverseWords(string s) {  
 string tmp, res;  
 for(int i = 0; i < s.size(); ++i){  
 if(s[i] == ' '){  
 if(tmp.size() > 0){  
 res = tmp + " " + res;  
 tmp = "";  
 }  
 }else{  
 tmp += s[i];  
 }  
 }  
 res = tmp + " " + res;  
 int i = 0, j = res.size() - 1;  
 while(i < s.size() && res[i] == ' ')i++;  
 while(j >= 0 && res[j] == ' ')j--;  
 return res.substr(i, j-i+1);  
 }  
};

**解法2** 字符串翻转。先将字符串整体翻转，然后将每个单词翻转。注意需要预处理除掉字符串首尾的空格，翻转结束后需要去除字符串中间多余的空格

class Solution {  
public:  
 string reverseWords(string s) {  
 while(s.size() > 0 && s[0] == ' ')s.erase(0,1);  
 while(s.size() > 0 && s[s.size()-1] == ' ')s.erase(s.size()-1, 1);  
 reverse(s, 0, s.size()-1);  
 int pre = 0, cur = 0;  
 while(cur < s.size()){  
 if(s[cur] == ' '){  
 reverse(s, pre, cur-1);  
 cur += 1;  
 pre = cur;  
 }else{  
 cur += 1;  
 }  
 }  
 reverse(s, pre, cur-1);  
 int i = 1;  
 while(i < s.size()){  
 if(s[i] == ' ' && s[i-1] == ' ')s.erase(i,1);  
 else i++;  
 }  
 return s;  
 }  
 void reverse(string &s, int pre, int cur){  
 while(pre < cur){  
 char tmp = s[pre];  
 s[pre] = s[cur];  
 s[cur] = tmp;  
 pre++;  
 cur--;  
 }  
 }  
};