Reverse Words in a String II

Given an input string, reverse the string word by word. A word is defined as a sequence of non-space characters.

The input string does not contain leading or trailing spaces and the words are always separated by a single space.

For example,
Given s = "`the sky is blue`",
return "`blue is sky the`".

Could you do it *in-place* without allocating extra space?

Related problem: Rotate Array

Solution 1

```
public void reverseWords(char[] s) {
    // Three step to reverse
    // 1, reverse the whole sentence
    reverse(s, 0, s.length - 1);
    // 2, reverse each word
    int start = 0;
    int end = -1;
    for (int i = 0; i < s.length; i++) {
        if (s[i] == ' ') {
            reverse(s, start, i - 1);
            start = i + 1;
        }
    }
    // 3, reverse the last word, if there is only one word this will solve the cor
ner case
    reverse(s, start, s.length - 1);
}

public void reverse(char[] s, int start, int end) {
    while (start < end) {
        char temp = s[start];
        s[start] = s[end];
        s[end] = temp;
        start++;
        end--;
    }
}
```

written by xzhu%40kabaminc.com original link here

Solution 2

```
void reverseWords(string &s) {
    reverse(s.begin(), s.end());
    for (int i = 0, j = 0; i < s.size(); i = j + 1) {
        for (j = i; j < s.size() && !isblank(s[j]); ++j);
        reverse(s.begin()+i, s.begin()+j);
    }
}
```

written by weibest original link here

Solution 3

```
class Solution {
public:
    void reverseWords(string &s) {
        vector<int> pos;
        pos.push_back(-1);
        for (int i = 0; i < s.size(); i++) if (s[i] == ' ') pos.push_back(i);
        pos.push_back(s.size());
        for (int i = pos.size() - 2; i >= 0; i--) reverse(s.begin() + pos[i] + 1,
s.begin() + pos[i + 1]);
        reverse(s.begin(), s.end());
    }
};
```

written by orzzzl original link here