

# Description

---

## 186. 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".

# Idea

---

We can create another temp array and assign the end of the input to the beginning of the temp array, and copy the reversed temp array back to the original array, this takes  $O(n)$  time and  $O(n)$  space. If we want to do it in-place, the operation we can use is swap. Because each word may have different length so it's hard to swap directly, so we can reverse the whole array first in order to make each word at the correct location without worrying about its length. Then we can reverse each word by word again to get the final reversed result.

Java

```
class Solution {
    public void reverseWords(char[] s) {
        reverse(s, 0, s.length - 1);
        int i = 0;
        int j = 0;
        while (j < s.length) {
            if (s[j] == ' ') {
                reverse(s, i, j - 1);
                i = j + 1;
            }
            j++;
        }
    }
}
```

```

    }
    reverse(s, i, j - 1);
}

private void reverse(char[] s, int i, int j) {
    while (i < j) {
        char t = s[i];
        s[i] = s[j];
        s[j] = t;
        i++;
        j--;
    }
}
}

```

C++

```

class Solution {
public:
    void reverseWords(vector<char>& str) {
        reverse(str, 0, str.size() - 1);
        int i = 0;
        int j = 0;
        while (j < str.size()) {
            if (str[j] == ' ') {
                reverse(str, i, j - 1);
                i = j + 1;
            }
            j++;
        }
        reverse(str, i, j - 1);
    }

    void reverse(vector<char>& str, int i, int j) {
        while (i < j) {
            char t = str[i];
            str[i] = str[j];
            str[j] = t;
            i++;
            j--;
        }
    }
};

```

## Summary

- 
- Array in-place operation use swap.