

Reverse Words in a String III

Given a string, you need to reverse the order of characters in each word within a sentence while still preserving whitespace and initial word order.

Example 1:

Input: "Let's take LeetCode contest"

Output: "s'teL ekat edoCteeL tsetnoc"

Note: In the string, each word is separated by single space and there will not be any extra space in the string.

Solution 1

C++

```
class Solution {
public:
    string reverseWords(string s) {
        for (int i = 0; i < s.length(); i++) {
            if (s[i] != ' ') { // when i is a non-space
                int j = i;
                for (; j < s.length() && s[j] != ' '; j++) { } // move j to the next space
                reverse(s.begin() + i, s.begin() + j);
                i = j - 1;
            }
        }

        return s;
    }
};
```

Java

```
public class Solution {
    public String reverseWords(String s) {
        char[] ca = s.toCharArray();
        for (int i = 0; i < ca.length; i++) {
            if (ca[i] != ' ') { // when i is a non-space
                int j = i;
                while (j + 1 < ca.length && ca[j + 1] != ' ') { j++; } // move j to the next space
                reverse(ca, i, j);
                i = j;
            }
        }
        return new String(ca);
    }

    private void reverse(char[] ca, int i, int j) {
        for (; i < j; i++, j--) {
            char tmp = ca[i];
            ca[i] = ca[j];
            ca[j] = tmp;
        }
    }
}
```

written by [alexander](#) original link [here](#)

Solution 2

Straightforward one.

```
class Solution(object):
    def reverseWords(self, s):
        """
        :type s: str
        :rtype: str
        """
        return " ".join(map(lambda x: x[::-1], s.split()))
```

written by [realisking](#) original link [here](#)

Solution 3

I just wanted to use the same logic as Reverse Words in a String II.

Step 1. Convert the string to char[] array

Step 2. Whenever I encounter a space ' ', I call the reverse function (just to keep the code clean)

Step 3. Repeat till the end!

Hope this helps! Thanks for voting :)

```
public String reverseWords(String s)
{
    char[] s1 = s.toCharArray();
    int i = 0;
    for(int j = 0; j < s1.length; j++)
    {
        if(s1[j] == ' ')
        {
            reverse(s1, i, j - 1);
            i = j + 1;
        }
    }
    reverse(s1, i, s1.length - 1);
    return new String(s1);
}

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

written by [sooryaprasanna](#) original link [here](#)

From [LeetCoder](#).