

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

Ruby:

Once again Ruby is really nice, it's super short and you can just write the steps from left to right. Split the string into words, reverse each word, then join them back together.

```
def reverse_words(s)
  s.split.map(&:reverse).join(" ")
end
```

Python:

Here I first reverse the order of the words and then reverse the entire string.

```
def reverseWords(self, s):
    return ' '.join(s.split()[::-1][::-1])
```

That's a bit shorter than the more obvious one:

```
def reverseWords(self, s):
    return ' '.join(x[::-1] for x in s.split())
```

Ruby again:

That double reversal in Ruby:

```
def reverse_words(s)
  s.split.reverse.join(" ").reverse
end
```

Python again:

The double reversal is not just shorter but also faster. Trying both versions as well as the optimized obvious solution (using a list comprehension instead of a generator expression), five attempts each:

```
>>> from timeit import timeit
>>> setup = 's = "Let\'s take LeetCode contest"'
>>> statements = (" ' '.join(s.split()[::-1])[::-1]",
                  "' '.join(x[::-1] for x in s.split())",
                  "' '.join([x[::-1] for x in s.split()])")
>>> for stmt in statements:
    print ' '.join('%0.2f' % timeit(stmt, setup) for _ in range(5)), 'seconds for: ', stmt

0.79 0.78 0.80 0.82 0.79 seconds for: ' '.join(s.split()[::-1])[::-1]
2.10 2.14 2.08 2.06 2.13 seconds for: ' '.join(x[::-1] for x in s.split())
1.27 1.26 1.28 1.28 1.26 seconds for: ' '.join([x[::-1] for x in s.split()])
```

With many more words, the double reversal's advantage gets even bigger:

```
>>> setup = 's = "Let\'s take LeetCode contest" * 1000'
>>> for stmt in statements:
    print ' '.join('%0.2f' % timeit(stmt, setup, number=1000) for _ in range(5))
, 'seconds for:', stmt

0.16 0.14 0.13 0.14 0.14 seconds for: ' '.join(s.split()[::-1])[::-1]
0.69 0.71 0.69 0.70 0.70 seconds for: ' '.join(x[::-1] for x in s.split())
0.63 0.68 0.63 0.64 0.64 seconds for: ' '.join([x[::-1] for x in s.split()])
```

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

Solution 2

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 end of the word
                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 3

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](#)

From [Leetcode](#).