186. Reverse Words in a String II

Jan. 19, 2020 | 12.9K views

≡ Articles > 186. Reverse Words in a String II ▼

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

Example:

```
Output: ["b","l","u","e"," ","i","s"," ","s","k","y"," ","t","h","e"]
```

Note:

- A word is defined as a sequence of non-space characters. • The input string does not contain leading or trailing spaces.
- The words are always separated by a single space.

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

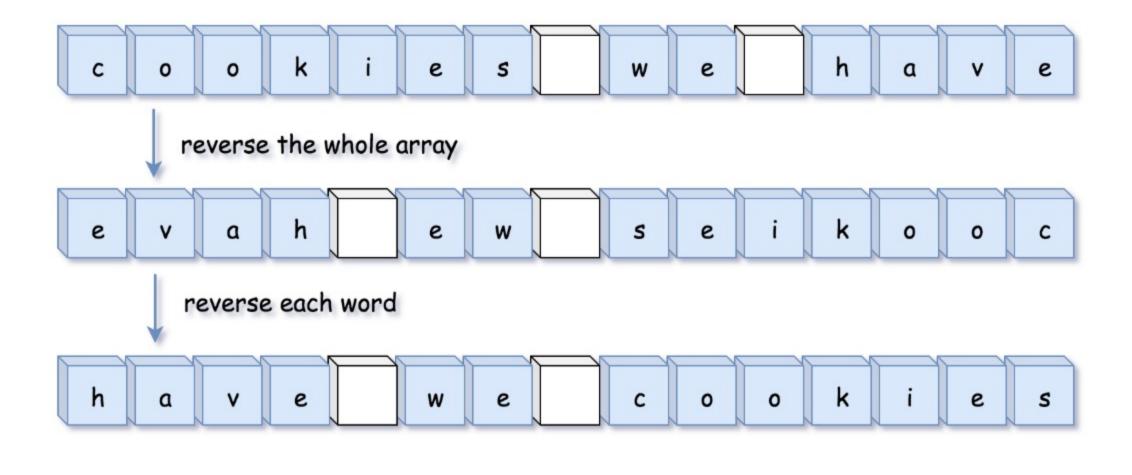
Solution

Approach 1: Reverse the Whole String and Then Reverse Each Word

one could aim $\mathcal{O}(1)$ space solution without any technical difficulties.

To have this problem in Amazon interview is a good situation, since input is a mutable structure and hence

The idea is simple: reverse the whole string and then reverse each word.



Let's first implement two functions:

Algorithm

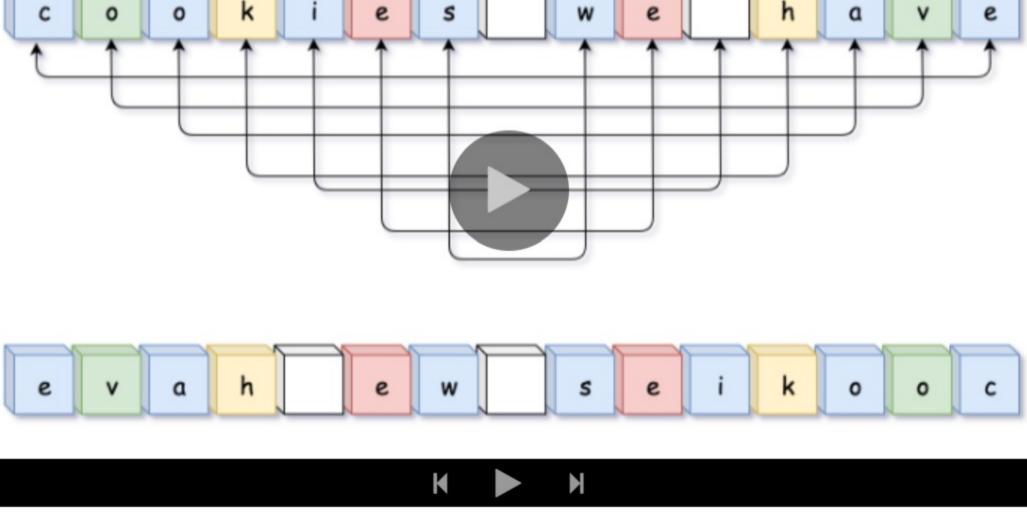
• reverse(1: list, left: int, right: int), which reverses array characters between left and

- right pointers. C++ users could directly use built-in std::reverse. • reverse_each_word(1: list), which uses two pointers to mark the boundaries of each word and
- previous function to reverse it. Now reverseWords(s: List[str]) implementation is straightforward:

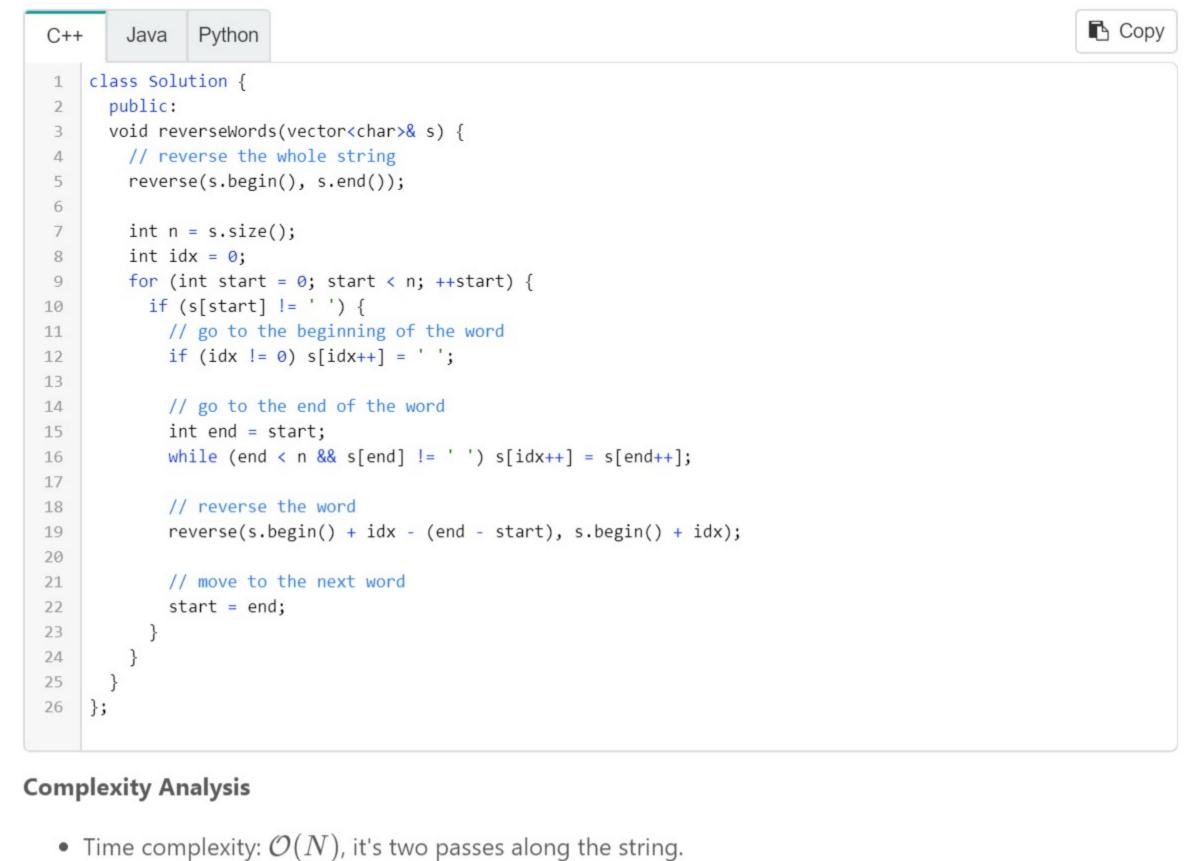
• Reverse the whole string: reverse(s, 0, len(s) - 1).

- Reverse each word: reverse_each_word(s).
- Implementation

1. Reverse the whole array using two-pointers swap

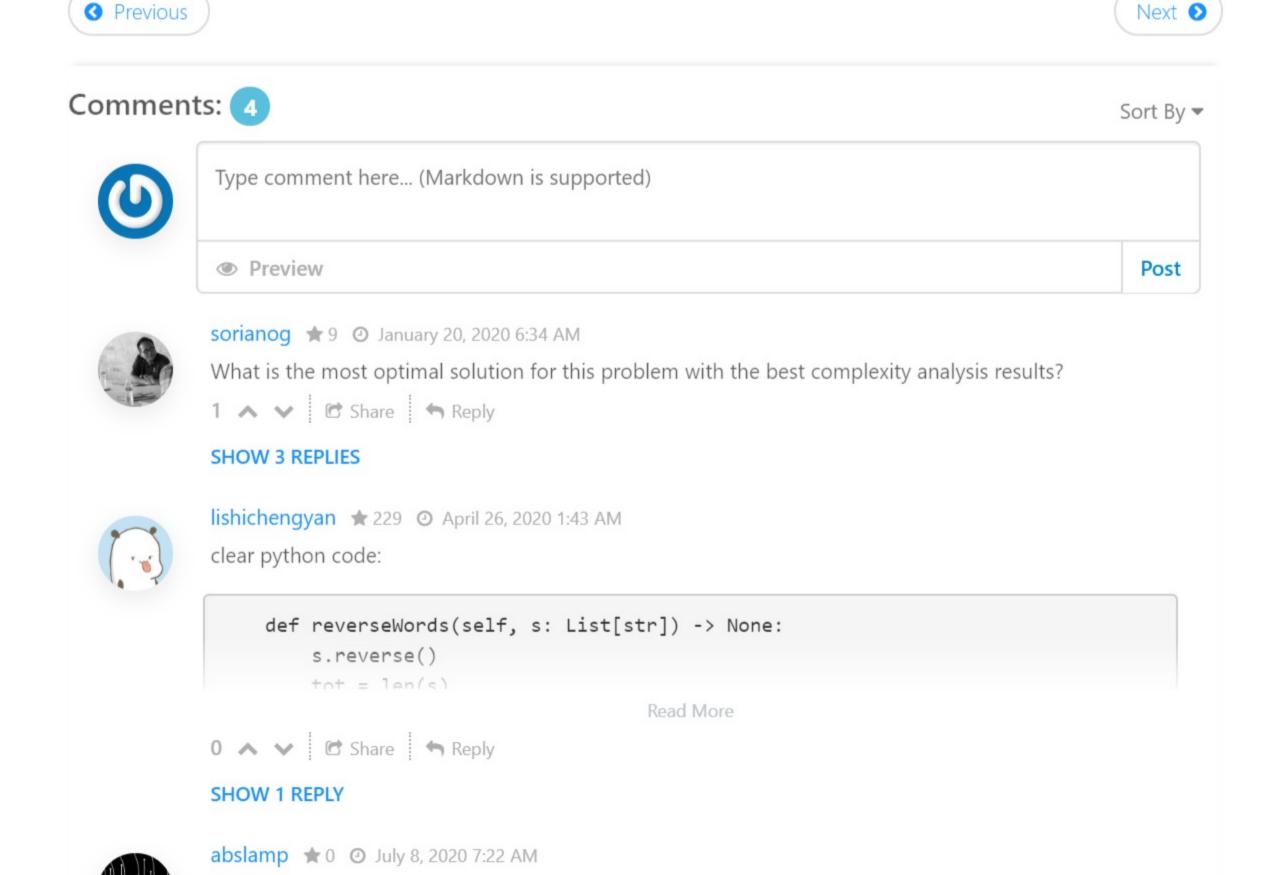


1/2



• Space complexity: $\mathcal{O}(1)$, it's a constant space solution.

- Rate this article: * * * *



Is that follow up (not taking any extra space) possible?

Maybe the follow up means "use constant space only"?

public String reverseWords(String text){

if(text == null | | text length() == A){

Read More

Isn't that two ints takes 2*sizeof(int) space?

cihandokur ★ 0 ② January 27, 2020 2:55 PM

0 ∧ ∨ ☑ Share ¬ Reply

another way