344. Reverse String ☐ (/problems/reverse-string/)

Oct. 13, 2019 | 175.6K views

Write a function that reverses a string. The input string is given as an array of characters `char[]` .

Do not allocate extra space for another array, you must do this by **modifying the input array in-place (https://en.wikipedia.org/wiki/In-place_algorithm)** with O(1) extra memory.

You may assume all the characters consist of printable ascii characters (https://en.wikipedia.org/wiki/ASCII#Printable_characters).

Example 1:

```
Input: ["h","e","l","l","o"]
Output: ["o","l","l","e","h"]
```

Example 2:

```
Input: ["H","a","n","n","a","h"]
Output: ["h","a","n","n","a","H"]
```

Solution

Overview

Life is short, use Python. (c)

```
Python                                                    📋 Copy
1  class Solution:
2      def reverseString(self, s):
3          s.reverse()
```

Speaking seriously, let's use this problem to discuss two things:

- Does *in-place* mean constant space complexity?

- Two pointers approach.

## Approach 1: Recursion, In-Place, $\mathcal{O}(N)$ Space

Does *in-place* mean constant space complexity?

No. By definition (https://en.wikipedia.org/wiki/In-place_algorithm), an in-place algorithm is an algorithm which transforms input using no auxiliary data structure.

The tricky part is that space is used by many actors, not only by data structures. The classical example is to use recursive function without any auxiliary data structures.

Is it in-place? Yes.

Is it constant space? No, because of recursion stack.
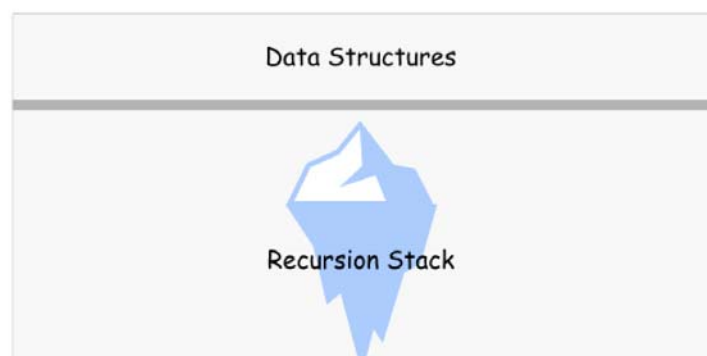
Algorithm

Here is an example. Let's implement recursive function `helper` which receives two pointers, left and right, as arguments.

- Base case: if `left >= right`, do nothing.

- Otherwise, swap `s[left]` and `s[right]` and call `helper(left + 1, right - 1)`.

To solve the problem, call helper function passing the head and tail indexes as arguments: `return helper(0, len(s) - 1)`.

Implementation

Java	Python		📋 Copy

```python
class Solution:
    def reverseString(self, s):
        def helper(left, right):
            if left < right:
                s[left], s[right] = s[right], s[left]
                helper(left + 1, right - 1)

        helper(0, len(s) - 1)
```

Complexity Analysis

- Time complexity : $\mathcal{O}(N)$ time to perform $N/2$ swaps.

- Space complexity : $\mathcal{O}(N)$ to keep the recursion stack.

Approach 2: Two Pointers, Iteration, $\mathcal{O}(1)$ Space

Two Pointers Approach

In this approach, two pointers are used to process two array elements at the same time. Usual implementation is to set one pointer in the beginning and one at the end and then to move them until they both meet.

Sometimes one needs to generalize this approach in order to use three pointers, like for classical Sort

Colors problem (https://leetcode.com/articles/sort-colors/).

Algorithm

- Set pointer left at index 0, and pointer right at index `n - 1`, where n is a number of elements in the array.

- While left < right:

    ○ Swap `s[left]` and `s[right]`.

    ○ Move left pointer one step right, and right pointer one step left.
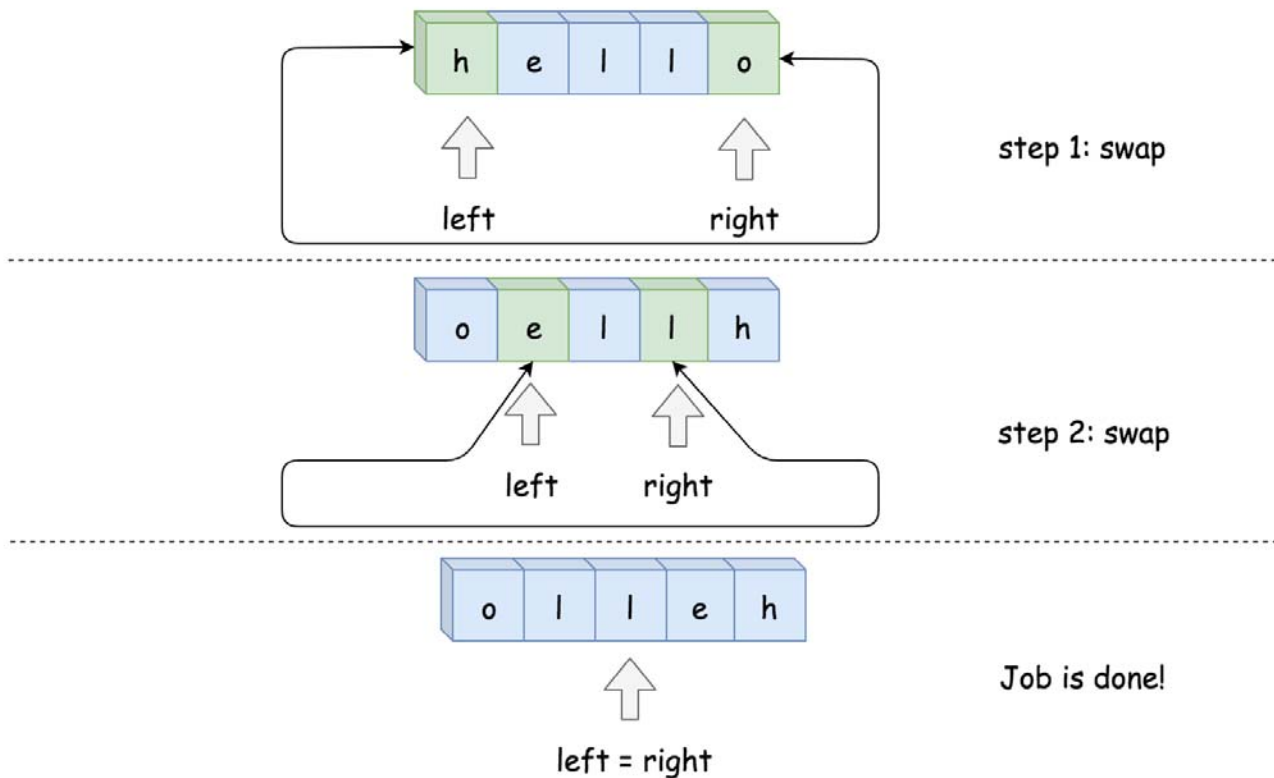
Implementation

```
Java    Python                                                          📋 Copy

1  class Solution:
2      def reverseString(self, s):
3          left, right = 0, len(s) - 1
4          while left < right:
5              s[left], s[right] = s[right], s[left]
6              left, right = left + 1, right - 1
```

Complexity Analysis

- Time complexity : $\mathcal{O}(N)$ to swap $N/2$ element.

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

Rate this article:

Comments: (43)

Sort By ▾

Type comment here... (Markdown is supported)

👁 Preview

Post

hyankov (/hyankov) ★ 174 ⏰ November 17, 2019 7:49 PM

Huh?! Why do we need two pointers again?

(/hyankov)

```
for (var i = 0; i < s.Length / 2; i++) {
    var tmp = s[i];
    s[i] = s[s.Length - i - 1];
```

Read More

62 ⌃ ⌄ ┃ ↪ Share ┃ ↩ Reply

SHOW 6 REPLIES

terrible_whiteboard (/terrible_whiteboard) ★ 586 ⏰ May 19, 2020 8:44 AM

I made a video if anyone is having trouble understanding the solution (clickable
link) https://youtu.be/uRk8ZIyMQkI (https://youtu.be/uRk8ZIyMQkI)

(/terrible_whiteboard)

Read More

19 ⌃ ⌄ ┃ ↪ Share ┃ ↩ Reply

thepatriot (/thepatriot) ★ 207 ⏰ November 13, 2019 4:07 PM

excellent article! short, sweet and clear. :)

(/thepatriot)

15 ⌃ ⌄ ┃ ↪ Share ┃ ↩ Reply

lord909 (/lord909) ★ 7 ⊙ December 26, 2019 2:43 PM

Why s[::-1] is not working ?

(/lord909)

6 ∧ ∨   ⧉ Share   ↰ Reply

SHOW 4 REPLIES

c_jain (/c_jain) ★ 19 ⊙ December 30, 2019 3:18 PM

This is the first problem for which I can say C++ one liner :)

```
for(int i=0; i<s.size()/2; i++) swap(s[i],s[s.size()-i-1]);
// or
reverse(s.begin(), s.end());
```

(/c_jain)

Read More

5 ∧ ∨ | ⧉ Share   ↰ Reply

SHOW 1 REPLY

heisenberg_blue (/heisenberg_blue) ★ 7 ⊙ January 17, 2020 10:34 AM

Can someone explain how the time complexity is O(N) even though we only loop n/2 times? Thanks.

4 ∧ ∨ | ⧉ Share   ↰ Reply

(/heisenberg_blue)

SHOW 5 REPLIES

osmankultur3 (/osmankultur3) ★ 9 ⊙ January 3, 2020 7:53 AM

OMG :D I literally had a lol moment when I saw the overview: Life is short, use Python. (c)

4 ∧ ∨ | ⧉ Share   ↰ Reply

(/osmankultur3)

josemb125 (/josemb125) ★ 2 ⊙ March 22, 2020 12:43 PM

hahaha life is short, use python to avoid covid-19

1 ∧ ∨ | ⧉ Share   ↰ Reply

(/josemb125)

heaventime (/heaventime) ★ 1 ⊙ March 5, 2020 5:49 AM

I get an error using this solution: 's' must consist of values from -2147483648 to 2147483647 only

1 ∧ ∨ | ⧉ Share   ↰ Reply

(/heaventime)

SHOW 3 REPLIES

andot (/andot) ★ 1 ⊙ February 15, 2020 12:34 AM

I get it, but the error given for submitting a non-O(1) answer should say so. Instead it was telling me the output was wrong, and I wasted an hour quadruple-checking that wasn't really the case.

1 ∧ ∨ | ⧉ Share   ↰ Reply

(/andot)

☰ Articles › 344. Reverse String ▾

Help Center (/support/) | Terms (/terms/) | Privacy (/privacy/)

United States (/region/)