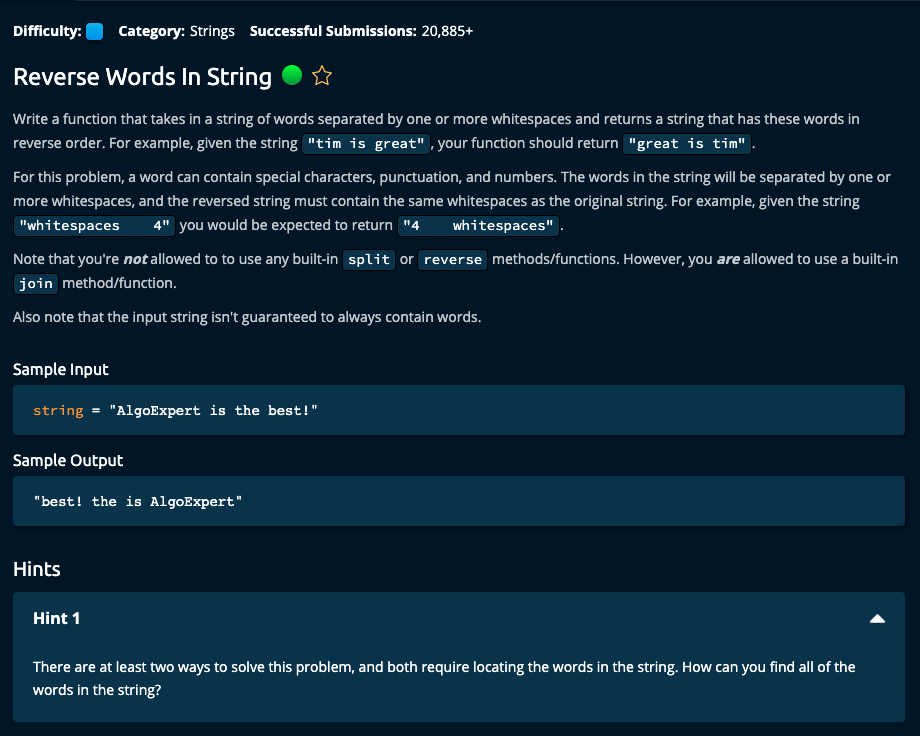
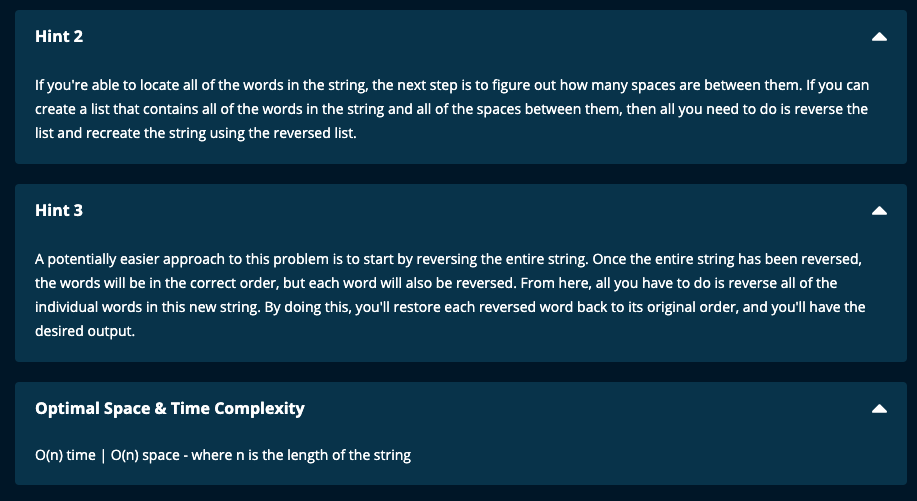
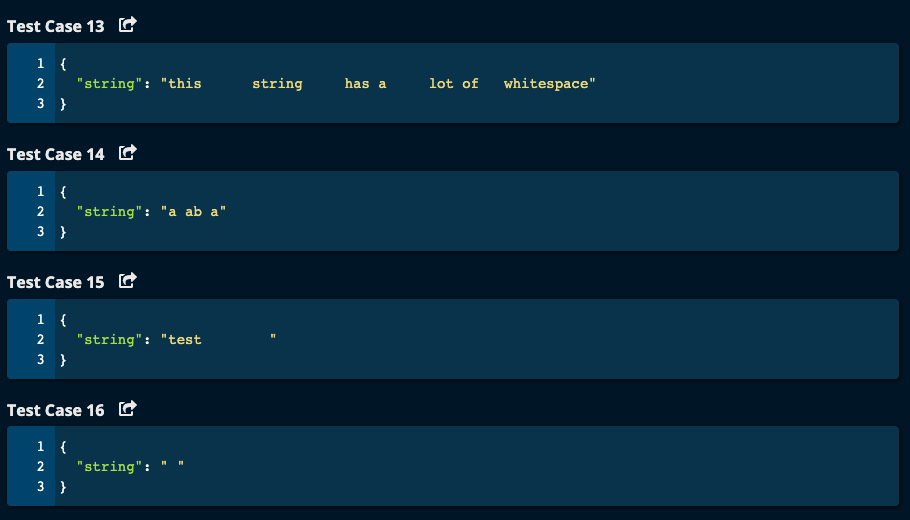
Reverse Words in String (Medium)







My Solutions:

Solution 1:

1. Initialize templist to an empty list and stack to an empty list.
2. Iterate through characters in the string. If the character is not a space (“ “), append the character to the templist.

If the character is a space, and if the templist is not empty, then join the characters in the templist to form a word, append the word to the stack and reinitialize the templist.

Append the space to the stack.

Note: This takes care of lot of whitespaces which are together.

1. At the end, for the last word i.e. if templist is not empty at the end of the iteration, we append it to the stack after converting templist to a string using join.
2. Initialize result to an empty list.
3. There are 2 ways:

First way is to pop the stack and get each word out of it. As we get each word, append the word to result. This is using pop() function which gives the last element.

Second way is to traverse the stack from the end in reverse order and append each word in the stack to result.

1. Finally join result (which is a list) to form a string and return the result as a string.
2. Time Complexity: O(n) | Space Complexity: O(n)

def reverseWordsInString(string):

alist = list(string)

templist = []

stack = []

for ch in string:

print("ch = ", ch)

if ch != ' ':

templist.append(ch)

else:

if templist:

word = ''.join(templist)

stack.append(word)

templist = []

stack.append(' ')

print("ch = ", ch, "templist = ", templist, "stack = ", stack)

if templist: # add the last word

stack.append(''.join(templist))

print("Final stack = ", stack)

result = []

'''

while stack:

word = stack.pop()

result.append(word)

return ''.join(result)

'''

for i in range(len(stack) - 1, -1, -1):

result.append(stack[i])

return "".join(result)

Solution 2:

1. Wrote a function called getReverse which takes a slice of a list and reverses the characters in the slice.
2. If the length of the string is less than or equal to 1, then simply return the string.
3. Make the string into a list and store it in result.
4. Reverse the characters in result.
5. Now we need to identify each word in result and reverse the individual words.
6. Initialize start to 0, and end to 0.
7. Iterate through result. If the current character is not a space, then increment end. If it is a space, then reverse the slice from start index to end index (note: end is not included in the slice). Then reinitialize start to end + 1, and end to start
8. For the last word which will remain as the last character may not be a space, the need to reverse the last slice of result.
9. Finally, join the elements of result to form a string and return it.
10. Time Complexity: O(n) | Space Complexity: O(n)

-----------------------------------

def reverseWordsInString(string):

if len(string) <= 1:

return string

result = [ ch for ch in string] # make the string into a list and store in result

getReverse(result, 0, len(string))

print("result = ", result)

start = 0 # Start of the word

end = 0 # End of the word

for i in range(len(string)):

if result[i] != " ":

end += 1

else:

getReverse(result, start, end)

start = end + 1

end = start

getReverse(result, start, end) # for the last word in the result list

print("result after reverse = ", result)

return "".join(result)

def getReverse(res, start, end):

'''

Take the slice res[start: end] and reverse the character between start and end

res = a list of characters

start = starting index

end = ending index

'''

left = start

right = end - 1

while left < right:

res[left], res[right] = res[right], res[left]

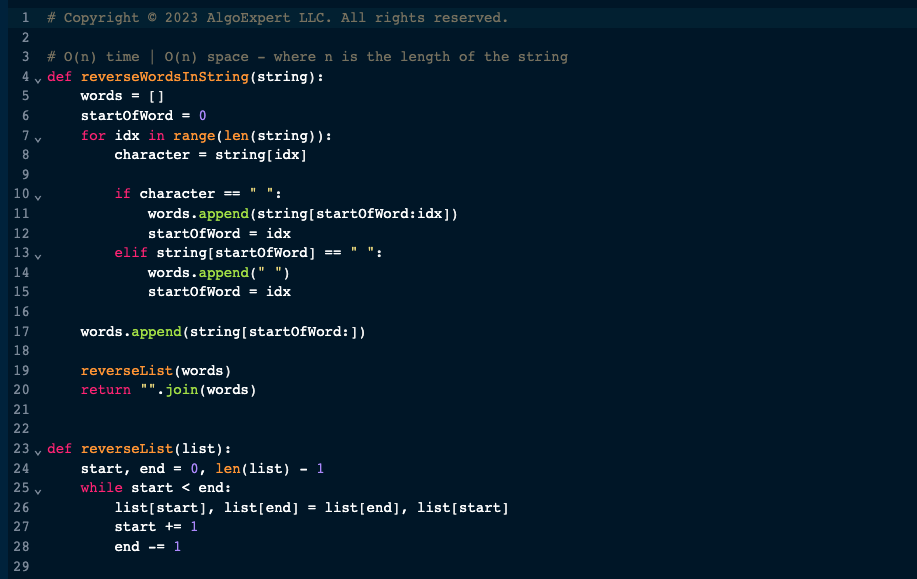
left += 1

right -= 1

=============================================

Algoexpert Solutions:

Solution 1:



1. Identify the words based on space and append them to words list.
2. The reverse each word in the word list using reverseList() function.

Solution 2:



1. Reverse the whole string.
2. Now each word in the string is reversed. Reverse each word one more time to get the original word.