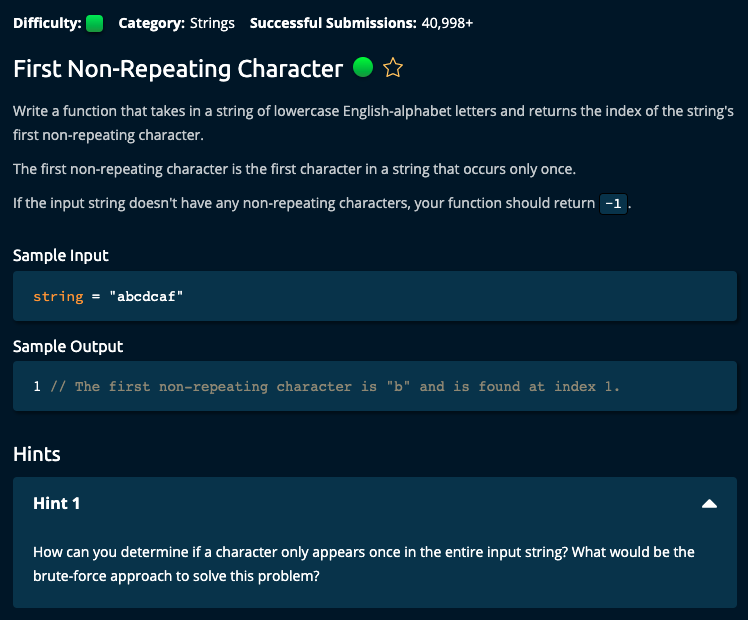
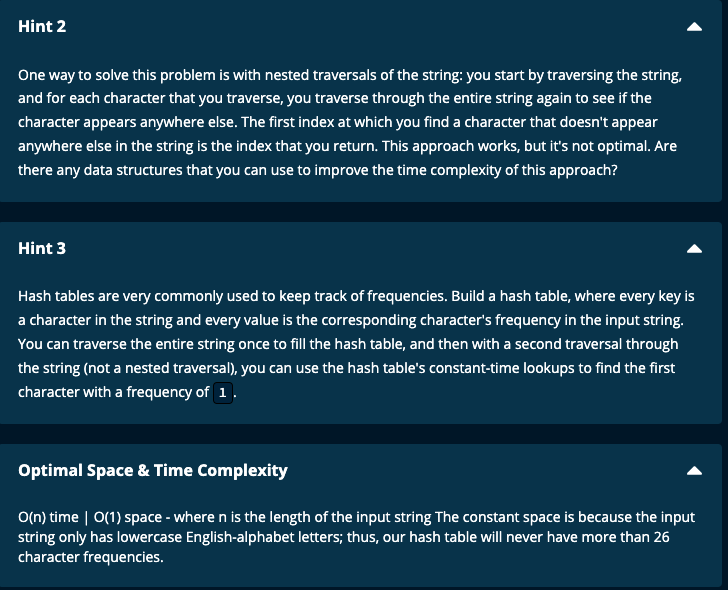
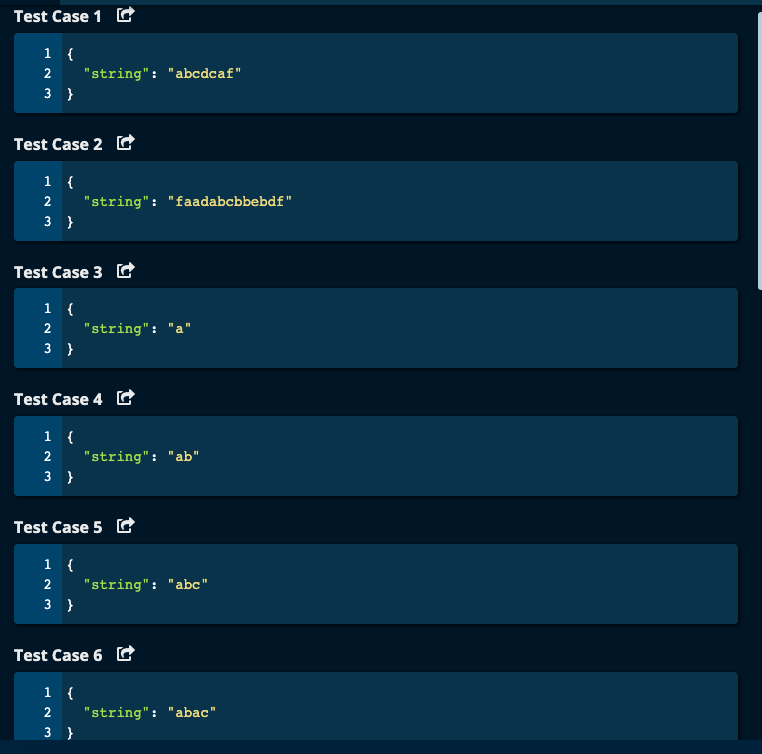
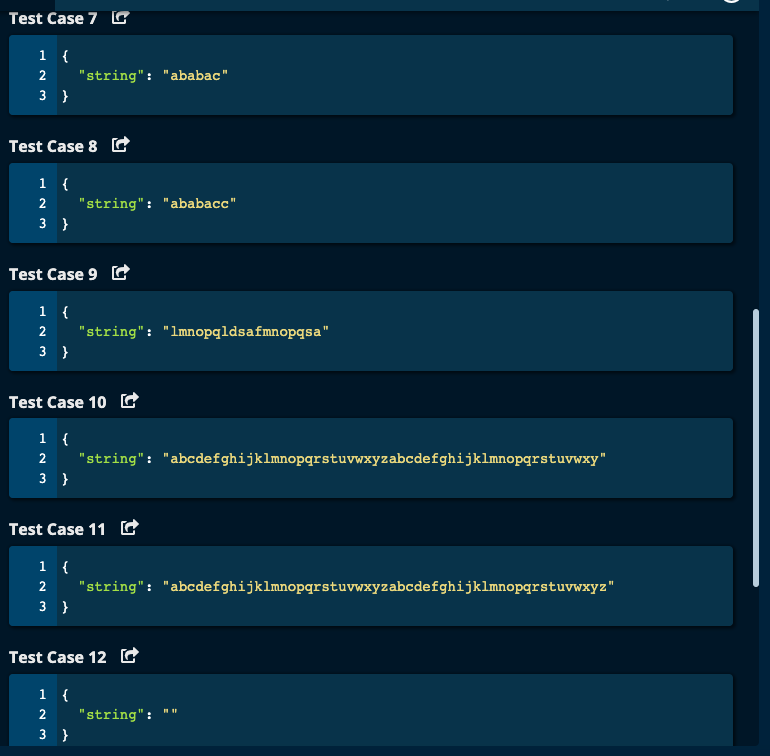
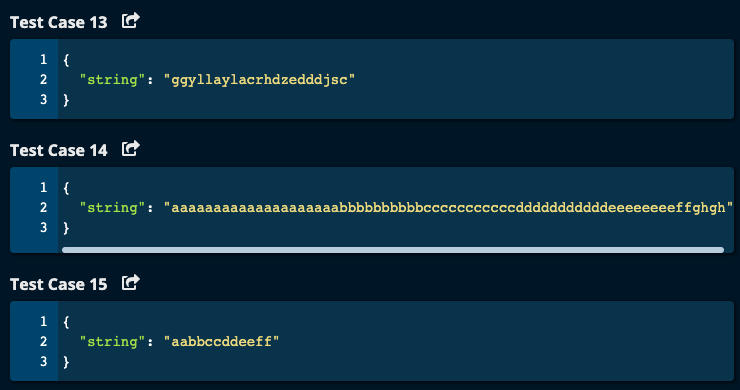
First Non-Repeating Character (Easy)











My Solution:

Solution 1:

def firstNonRepeatingCharacter(string):

charDict = {}

for ch in string:

if ch not in charDict:

charDict[ch] = 1

else:

charDict[ch] += 1

for i, ch in enumerate(string):

if charDict[ch] == 1:

return i

return -1

JJ Notes:

1. Initialize charDict to an empty dictionary to store the frequency dictionary.
2. Iterate through the string and store the frequency of characters in the string.
3. Iterate through the string a second time and when you find a character that has a frequency of 1 in the dictionary, then return its index.
4. Return -1 if there is no character with a frequency of 1.
5. Time complexity: O(n) since we iterate 2 times through the string at the most.

Space complexity: O(1) since there are only 26 lower cases alphabets and so dictionary contains at most 26 keys.

Solution 2:

from collections import Counter

def firstNonRepeatingCharacter(string):

charDict = Counter(string)

for i, ch in enumerate(string):

if charDict[ch] == 1:

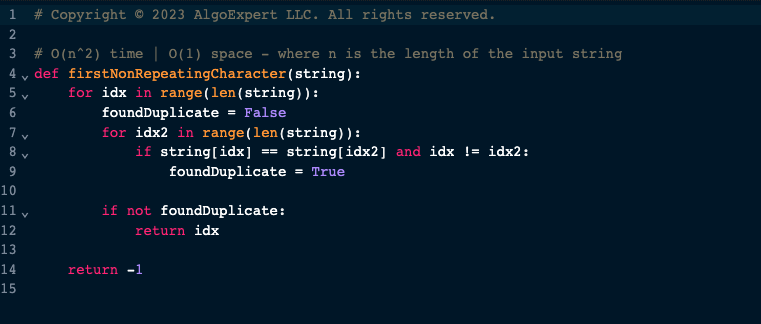
return i

return -1

JJ Notes: To get frequency dictionary use collections and import Counter. Otherwise same as Solution 1.

Algoexpert Solutions:

Solution 1: Brute Force



Solution 2:

JJ Notes:

Very similar to my solution

