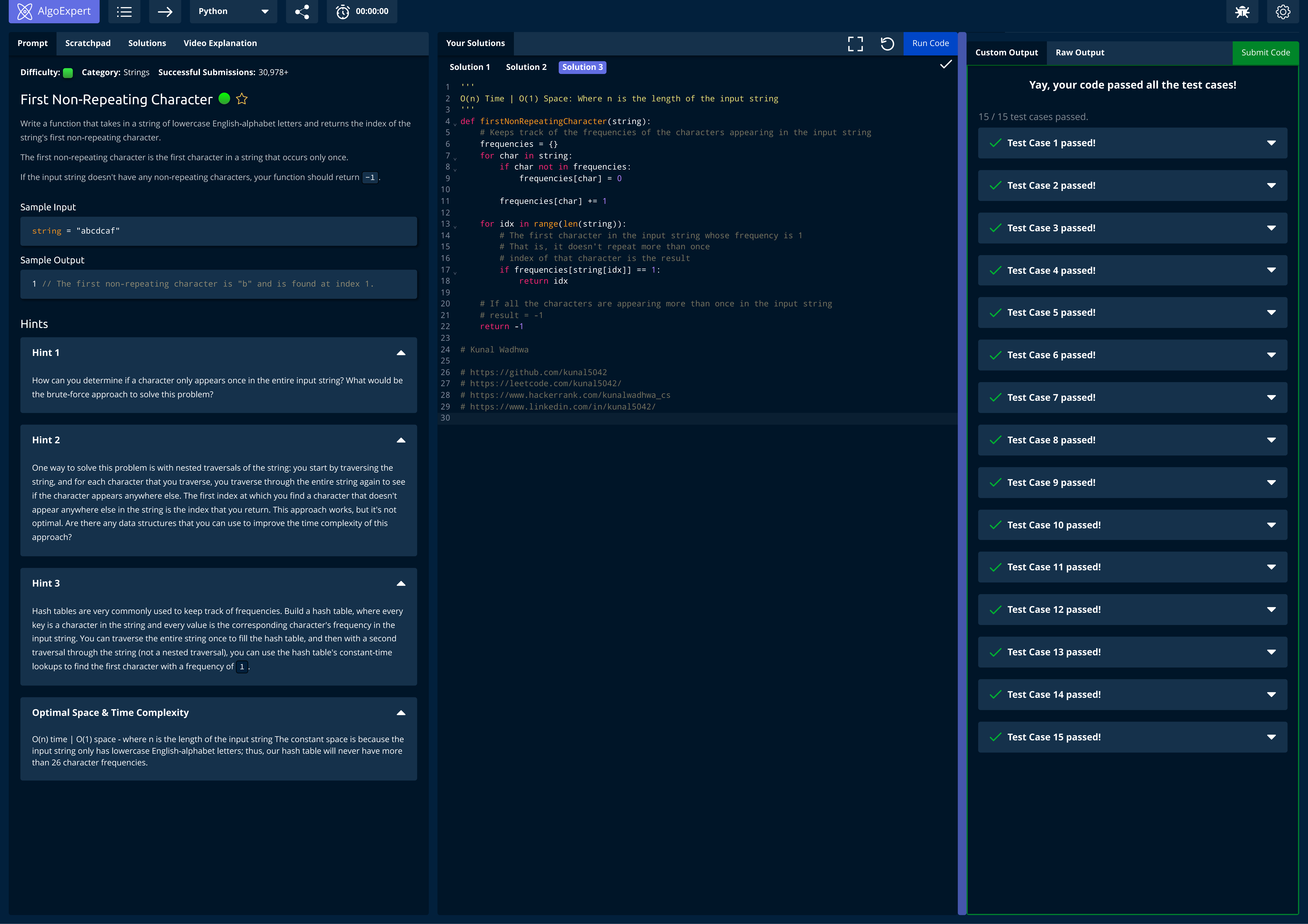

Difficulty: 🟩 **Category:** Strings **Successful Submissions:** 30,978+

First Non-Repeating Character ● ☆

Write a function that takes in a string of lowercase English-alphabet letters and returns the index of the string's first non-repeating character.

The first non-repeating character is the first character in a string that occurs only once.

If the input string doesn't have any non-repeating characters, your function should return `-1`.

Sample Input

```
string = "abcdcaf"
```

Sample Output

```
1 // The first non-repeating character is "b" and is found at index 1.
```

Hints

Hint 1

How can you determine if a character only appears once in the entire input string? What would be the brute-force approach to solve this problem?

Hint 2

One way to solve this problem is with nested traversals of the string: you start by traversing the string, and for each character that you traverse, you traverse through the entire string again to see if the character appears anywhere else. The first index at which you find a character that doesn't appear anywhere else in the string is the index that you return. This approach works, but it's not optimal. Are there any data structures that you can use to improve the time complexity of this approach?

Hint 3

Hash tables are very commonly used to keep track of frequencies. Build a hash table, where every key is a character in the string and every value is the corresponding character's frequency in the input string. You can traverse the entire string once to fill the hash table, and then with a second traversal through the string (not a nested traversal), you can use the hash table's constant-time lookups to find the first character with a frequency of `1`.

Optimal Space & Time Complexity

O(n) time | O(1) space - where n is the length of the input string The constant space is because the input string only has lowercase English-alphabet letters; thus, our hash table will never have more than 26 character frequencies.

```python
'''
O(n) Time | O(1) Space: Where n is the length of the input string
'''
def firstNonRepeatingCharacter(string):
    # Keeps track of the frequencies of the characters appearing in the input string
    frequencies = {}
    for char in string:
        if char not in frequencies:
            frequencies[char] = 0

        frequencies[char] += 1

    for idx in range(len(string)):
        # The first character in the input string whose frequency is 1
        # That is, it doesn't repeat more than once
        # index of that character is the result
        if frequencies[string[idx]] == 1:
            return idx

    # If all the characters are appearing more than once in the input string
    # result = -1
    return -1

# Kunal Wadhwa

# https://github.com/kunal5042
# https://leetcode.com/kunal5042/
# https://www.hackerrank.com/kunalwadhwa_cs
# https://www.linkedin.com/in/kunal5042/

```

Yay, your code passed all the test cases!

15 / 15 test cases passed.

✓ **Test Case 1 passed!**

✓ **Test Case 2 passed!**

✓ **Test Case 3 passed!**

✓ **Test Case 4 passed!**

✓ **Test Case 5 passed!**

✓ **Test Case 6 passed!**

✓ **Test Case 7 passed!**

✓ **Test Case 8 passed!**

✓ **Test Case 9 passed!**

✓ **Test Case 10 passed!**

✓ **Test Case 11 passed!**

✓ **Test Case 12 passed!**

✓ **Test Case 13 passed!**

✓ **Test Case 14 passed!**

✓ **Test Case 15 passed!**