Difficulty: **Category:** Strings **Successful Submissions:** 18,081+

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.