HATFD1015

Problem Statement

Find the First Non-Repeating Character
Write a program to find the first non-repeating character in a string. For input `"swiss"`, the output should be `"w"`. You cannot use any built-in string or character frequency counting functions.

Program:

```
import java.util.*;

class Main {

    // Method to find the First Non-Repeating Character in a String
    public static Character findNonRepeating(String s){

        // Creating the HashMap to store the frequencies of each character
        HashMap<Character, Integer> map = new HashMap<>();

        // Traversing each character in a string and store its frequencies
        for(int i=0; i<s.length(); i++){
            map.put(s.charAt(i), map.getOrDefault(s.charAt(i), 0) + 1);
        }

        // Again Traversing each character in the string to find the first non-repeating character
        for(char c : s.toCharArray()){
            if(map.get(c) == 1){
                return c;
            }
        }

        // Returning null if no non-repeating characters found
        return null;
    }


    public static void main(String[] args) {

        // Testing the findNonRepeating method with different strings
        char c1 = findNonRepeating("abcdef");
        System.out.println(c1);

        char c2 = findNonRepeating("apple");
        System.out.println(c2);

        char c3 = findNonRepeating("aabbccdd");
        System.out.println(c3);
    }
}
```

Performance:

Time Complexity: O(n + n) = O(n), where n is the length of the string**.**
Space Complexity: O(c), where c is the number of unique characters in a string that is stored in the HashMap.

Testcases:

Input 1:

S = “abcdef”
Output = ‘a’.

Input 2:

S = “apple”
Output = ‘a’.

Input 3:

S = “aabbccdd”
Output = null.