PRACTICE COMPETE JOBS LEADERBOARD Search svolpe

All Contests > USC ACM Fall 2018 350+ Division > Biskit's Brain Games

Biskit's Brain Games

locked

by vmcquinn

Problem | Submissions | Leaderboard | Discussions

Submitted 6 days ago • Score: 30.00

Status: Accepted

✓ Test Case #0 ✓ Test Case #1 ✓ Test Case #2
✓ Test Case #3 ✓ Test Case #4 ✓ Test Case #5
✓ Test Case #6 ✓ Test Case #7 ✓ Test Case #8
✓ Test Case #9 ✓ Test Case #10 ✓ Test Case #11
✓ Test Case #12 ✓ Test Case #13 ✓ Test Case #14
✓ Test Case #15

Submitted Code

Language: Java 8

Fork into a snippet Open in editor

```java
import java.io.*;
import java.lang.*;
import java.util.*;
public class Solution{
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        String a = keyboard.nextLine();
        a = a.toLowerCase();
        String b = keyboard.nextLine();
        b = b.toLowerCase();
        Map<Character,Integer> charCounter = new HashMap<Character,Integer>();
        Map<Character,Integer> charCounter1 = new HashMap<Character,Integer>();
        //Populate hashmap
        for(int i = 0; i < a.length(); i++){
            charCounter.put(a.charAt(i), charCounter.getOrDefault(a.charAt(i), 0)+1);
        }
        for(int i = 0; i < b.length(); i++){
            charCounter1.put(b.charAt(i), charCounter1.getOrDefault(b.charAt(i), 0)+1);
        }
        //Get the difference between the two hashmaps
        int [] counts = new int [26];
        for(int i = 0; i < counts.length; i++){
            char ch = (char)(97 + i);
            counts[i] += charCounter.getOrDefault(ch,0) - charCounter1.getOrDefault(ch,0);
        }
        int total = 0;
        for(int i = counts.length-1; i >= 0; i--){
            int neighbor = i-1; // num is to the left of our current character
            int current = i;
            if(counts[current] == 0){
                continue;
            }
                // If counts[i] is positive we need to add the string to b
                // If counts[i] is negative we need to add the string to a
                while(counts[current] != 0 && neighbor >= 0){
                    if(counts[neighbor] < 0 && counts[current] > 0){
                        int min = Math.min(counts[current],Math.abs(counts[neighbor]));
                        counts[current] -= min;
                        counts[neighbor] += min;
                        total += (current - neighbor) * min;
                    }
                    if(counts[neighbor] > 0 && counts[current] < 0){
                        int min = Math.min(Math.abs(counts[current]),counts[neighbor]);
                        counts[current] += min;
                        counts[neighbor] -= min;
                        total += (current - neighbor) * min;
                    }
                    neighbor --;
                }
                if(neighbor == -1 && counts[current] != 0){
                    total += Math.abs(counts[current]) * (current+1);
                }
        }
        // for(int i = 0; i < counts.length; i++){
        //     System.out.println(counts[i]);
        // }
        if (total == 0) {
            System.out.println("AC JUMBLE");
        } else {
            System.out.println(total);
        }
    }
}
```

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature