Practice Compete Jobs Rank Leaderboard 1 nachomonllor

All Domains > Algorithms > Strings > Anagram

Badge Progress (Details)

Points: 1250.76 Rank: 14830

Anagram

by amititkgp

Problem Submissions Leaderboard Discussions Editorial Topics

Sid is obsessed with reading short stories. Being a CS student, he is doing some interesting frequency analysis with the books. He chooses strings $S1$ and $S2$ in such a way that $|len(S1) - len(S2)| \leq 1$.

Your task is to help him find the minimum number of characters of the first string he needs to change to enable him to make it an anagram of the second string.

Note: A word *x* is an anagram of another word *y* if we can produce *y* by rearranging the letters of *x*.

Input Format

The first line will contain an integer, T, representing the number of test cases. Each test case will contain a string having length $len(S1) + len(S2)$, which will be concatenation of both the strings described above in the problem.The given string will contain only characters from a to z.

Constraints

- $1 \leq T \leq 100$
- $1 \leq len(S1) + len(S2) \leq 10^4$

Output Format

An integer corresponding to each test case is printed in a different line, i.e. the number of changes required for each test case. Print -1 if it is not possible.

Sample Input

```
6
aaabbb
ab
abc
mnop
xyyx
xaxbbbxx
```

Sample Output

```
3
1
-1
2
0
1
```

Explanation

Test Case #01: We have to replace all three characters from the first string to make both of strings anagram. Here, $S1$ = "aaa" and $S2$ = "bbb". So the solution is to replace all character 'a' in string *a* with character 'b'.

Test Case #02: You have to replace 'a' with 'b', which will generate "bb".

Test Case #03: It is not possible for two strings of unequal length to be anagram for each other.

Test Case #04: We have to replace both the characters of first string ("mn") to make it anagram of other one.

Test Case #05: $S1$ and $S2$ are already anagram to each other.

Test Case #06: Here *S1* = "*xaxb*" and *S2* = "*bbxx*". He had to replace 'a' from *S1* with 'b' so that *S1* = "*xbxb*" and we can rearrange its letter to "bbxx" in order to get *S2*.

Submissions: 29808
Max Score: 25
Difficulty: Easy

Rate This Challenge:
☆☆☆☆☆

Need Help?
String Basics

Anagram

More

Current Buffer (saved locally, editable)

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
class Solution {


    static void Main(string[] args)
        {

            int t = int.Parse(Console.ReadLine());

            while (t-- > 0)
            {

                //string s = "aaabbb";
                //string s = "mnop";
                //string s = "xaxbbbxx";
                string s = Console.ReadLine();

                if (s.Length % 2 != 0)
                {
                    Console.WriteLine(-1);
                      continue;
                }

                string a = s.Substring(0, s.Length / 2);
                string b = s.Substring(s.Length / 2, s.Length / 2);
                //Console.WriteLine(a);
                //Console.WriteLine(b);


                Dictionary<char, int> frec_a = a.ToCharArray().GroupBy(x => x)
                                       .ToDictionary(x => x.Key, x => x.Count());
                Dictionary<char, int> frec_b = b.ToCharArray().GroupBy(x => x)
                                   .ToDictionary(x => x.Key, x => x.Count());

                int sobra_a = 0, sobra_b = 0;

                foreach (KeyValuePair<char, int> kvp in frec_a)
                {
                    if (frec_b.ContainsKey(kvp.Key))
                    {
                        if (kvp.Value >= frec_b[kvp.Key])
                        {
                            sobra_a += kvp.Value - frec_b[kvp.Key];
                        }
                        else
                        {
                            sobra_b += frec_b[kvp.Key] - kvp.Value;
                        }
                    }
                    else
                    {
                        sobra_a += kvp.Value;
                    }
                }
                foreach (KeyValuePair<char, int> kvp in frec_b)
                {
                    if (!frec_a.ContainsKey(kvp.Key))
                    {
                        sobra_b += kvp.Value;
                    }
                }

                Console.WriteLine(Math.Min(sobra_a, sobra_b));

            }


           // Console.ReadLine();
        }



}
```

Line: 70 Col: 1

Upload Code as File □ Test against custom input

Run Code Submit Code

Congrats, you solved this challenge!

✓ 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

Next Challenge

Join us on IRC at #hackerrank on freenode for hugs or bugs.

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