Making Anagrams

by amititkgp

Problem | Submissions | Leaderboard | Discussions | Editorial | Topics

Alice is taking a cryptography class and finding *anagrams* to be very useful. We consider two strings to be anagrams of each other if the first string's letters can be rearranged to form the second string. In other words, both strings must contain the same exact letters in the same exact frequency For example, `bacdc` and `dcbac` are anagrams, but `bacdc` and `dcbad` are not.

Alice decides on an encryption scheme involving two large strings where encryption is dependent on the minimum number of character deletions required to make the two strings anagrams. Can you help her find this number?

Given two strings, a and b, that may or may not be of the same length, determine the minimum number of character deletions required to make a and b anagrams. Any characters can be deleted from either of the strings.

This challenge is also available in the following translations:

- Chinese
- Russian

Input Format

The first line contains a single string, a.
The second line contains a single string, b.

Constraints

- $1 \leq |a|, |b| \leq 10^4$
- It is guaranteed that a and b consist of lowercase English letters.

Output Format

Print a single integer denoting the number of characters which must be deleted to make the two strings anagrams of each other.

Sample Input

```
cde
abc
```

Sample Output

```
4
```

Explanation

We delete the following characters from our two strings to turn them into anagrams of each other:

1. Remove `d` and `e` from `cde` to get `c`.
2. Remove `a` and `b` from `abc` to get `c`.

We had to delete 4 characters to make both strings anagrams, so we print 4 on a new line.

Submissions: 28933
Max Score: 30
Difficulty: Easy

Rate This Challenge:

Need Help?

String Basics

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) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution */

        string a = Console.ReadLine();
        string b = Console.ReadLine();

        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( sobra_a + sobra_b );


    }
}
```

Line: 4 Col: 19

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