Given two strings, your task is to find out if there is a *swap sequence* that transforms the first string into the second string. A *swap sequence* is a sequence of strings, such that for each two consecutive string the one can be obtained from another after a swap of two characters in one of the strings.

Return true if a swap sequence exists andfalse otherwise.

**Example**

* swapSequence("abcd", "dacb") = true.

There is a *swap sequence* that transforms"abcd" into "dacb". For example, this sequence can be: "abcd" -> "adcb" -> "dacb". First, you swap the characters 'b'and 'd', and after this, you swap the characters 'a and 'd'.

* swapSequence("abcd","abab") = false

There is no *swap sequence* that transforms"abcd" into "abab".

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] string a**

The first string, a string of lowercase English letters.

*Constraints:*  
1 ≤ a.length ≤ 20.

* **[input] string b**

The second string, a string of lowercase English letters.

*Constraints:*  
b.length = a.length.

* **[output] boolean**

true if there is a *swap sequence*, andfalse otherwise.

<https://codefights.com/challenge/ZYcY8u59FAog4F5fk/main>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static bool swapSequence(string a, string b)

{

Dictionary<char, int> da = new Dictionary<char, int>();

foreach (char ch in a)

{

if (da.ContainsKey(ch))

{

da[ch]++;

}

else

{

da[ch] = 1;

}

}

Dictionary<char, int> db = new Dictionary<char, int>();

foreach (char ch in b)

{

if (db.ContainsKey(ch))

{

db[ch]++;

}

else

{

db[ch] = 1;

}

}

foreach (KeyValuePair<char, int> kvp in da)

{

if (!db.ContainsKey(kvp.Key))

{

return false;

}

else

{

if (kvp.Value != db[kvp.Key])

{

return false;

}

}

}

return true;

}

static void Main(string[] args)

{

Console.WriteLine(swapSequence("abcdefghijklmnopqrst", "tsrqponmlkjihgfedcba"));

Console.ReadLine();

}

}

}