Common Child

by naonaonao

Given two strings a and b of equal length, what's the longest string (s) that can be constructed such that it is a child of both?

A string x is said to be a child of a string y if x can be formed by deleting 0 or more characters from y.

For example, `ABCD` and `ABDC` has two children with maximum length 3, `ABC` and `ABD`. Note that we will not consider `ABCD` as a common child because `C` doesn't occur before `D` in the second string.

Input format

Two strings, a and b, with a newline separating them.

Constraints

- $1 \leq |a|, |b| \leq 5000$
- All characters are upper cased and lie between ASCII values 65-90.

Output format

Print the length of the longest string s, such that s is a child of both a and b.

Sample Input 0

```
HARRY
SALLY
```

Sample Output 0

```
2
```

The longest possible string that is possible by deleting zero or more characters from $HARRY$ and $SALLY$ is AY, whose length is 2.

Sample Input 1

```
AA
BB
```

Sample Output 1

```
0
```

AA and BB has no characters in common and hence the output is 0.

Sample Input 2

```
SHINCHAN
NOHARAAA
```

Sample Output 2

```
3
```

The longest string that can be formed between $SHINCHAN$ and $NOHARAAA$ while maintaining the order is NHA.

Sample Input 3

```
ABCDEF
FBDAMN
```

Sample Output 3

```
2
```

BD is the longest child of the given strings.

Submissions: 14893
Max Score: 60
Difficulty: Medium

Rate This Challenge:

Need Help?

Dynamic Programming Basics

More

Current Buffer (saved locally, editable) Java 8

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    static int commonChild(String str1, String str2){

        int[][] arr = new int[str1.length()][str2.length()];

        for(int j = 0 ; j < str1.length() ; j++){

            for(int k = 0 ; k < str2.length() ; k++){

                if(k == 0 && j == 0 && str2.charAt(k) == str1.charAt(j)){
                    arr[j][k] = 1;
                }
                else if(k == 0 && j == 0 && str2.charAt(k) != str1.charAt(j)){
                    arr[j][k] = 0;
                }
                else if(k == 0){
                    if(str1.charAt(j) == str2.charAt(k)){
                        arr[j][k] = 1;
                    }
                    else{
                        arr[j][k] = Math.max(arr[j-1][k],0);
```

```
                    }
                }
                else if(j == 0){
                    if(str1.charAt(j) == str2.charAt(k)){
                        arr[j][k] = 1;
                    }
                    else{
                        arr[j][k] = Math.max(0,arr[j][k-1]);
                    }
                }
                else if(str2.charAt(k) == str1.charAt(j)){
                    arr[j][k] = arr[j - 1][k - 1] + 1;
                }
                else if(str2.charAt(k) != str1.charAt(j)){
                    arr[j][k] = Math.max(arr[j - 1][k],Math.max(arr[j - 1][k - 1],arr[j][k - 1]));
                }
            }
        }

        return arr[str1.length() - 1][str2.length() - 1];

    }

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String s1 = in.next();
        String s2 = in.next();
        int result = commonChild(s1, s2);
        System.out.println(result);
    }
}

```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code