Thursday, December 16, 2021 11:59 AM

Transform String

{ Medium }

Given two strings A and B. Find the minimum number of steps required to transform string A into string B. The only allowed operation for the transformation is selecting a character from string A and inserting it in the beginning of string A.

From <https://practice.geeksforgeeks.org/problems/transform-string5648/1>

Example 1:

Input:
A = "abd"
B = "bad"
Output: 1
Explanation: The conversion can take place in
1 operation: Pick 'b' and place it at the front.

From <https://practice.geeksforgeeks.org/problems/transform-string5648/1>

Example 2:

Input:
A = "GeeksForGeeks"
B = "ForGeeksGeeks"
Output: 3

Explanation: The conversion can take
place in 3 operations:
Pick 'r' and place it at the front.
A = "rGeeksFoGeeks"
Pick 'o' and place it at the front.
A = "orGeeksFGeeks"
Pick 'F' and place it at the front.
A = "ForGeeksGeeks"

From <https://practice.geeksforgeeks.org/problems/transform-string5648/1>

Your Task:

You dont need to read input or print anything. Complete the function **transform()** which takes two strings A and B as input

parameters and returns the minimum number of steps required to transform A into B. If transformation is not possible return -1.

Expected Time Complexity: O(N) where N is max(length of A, length of B)

Expected Auxiliary Space: O(1)

From <https://practice.geeksforgeeks.org/problems/transform-string5648/1>

Company Tags

- **Directi**

Topic Tags

- **Strings**
- **Hash**
- **Map**

From <https://practice.geeksforgeeks.org/problems/transform-string5648/1>

From <https://practice.geeksforgeeks.org/problems/transform-string5648/1>

HINT

To find minimum number of operations, start matching from last characters of both strings.

If last characters match, then our task reduces to n-1 characters.

If last characters don't match, then find the position of B's mismatching character in A.

The difference between two positions indicates that these many characters of A must be moved before current character of A.

From <https://practice.geeksforgeeks.org/problems/transform-string5648/1#>

Below is complete algorithm.

1) Find if A can be transformed to B or not by first creating a count array for all characters of A, then checking with B if B has same count for every character.

2) Initialize result as 0.

3) Start traversing from end of both strings.

……a) If current characters of A and B match, i.e., A[i] == B[j]

……then do i = i-1 and j = j-1

b) If current characters don't match, then search B[j] in remaining

……A. While searching, keep incrementing result as these characters

……must be moved ahead for A to B transformation.

From <https://www.geeksforgeeks.org/transform-one-string-to-another-using-minimum-number-of-given-operation/>

```java
// Java program to find minimum number of
// operations required to transform one
// string to other
import java.io.*;
import java.util.*;

public class GFG {

    // Function to find minimum number of
    // operations required to transform
    // A to B.
    public static int minOps(String A, String B)
    {

        // This parts checks whether conversion is
        // possible or not
        if(A.length() != B.length())
            return -1;

        int i, j, res = 0;
        int count [] = new int [256];

        // count characters in A

        // subtract count for every character in B
        for(i = 0; i < A.length(); i++)
        {
            count[A.charAt(i)]++;
            count[B.charAt(i)]--;
        }

        // Check if all counts become 0
        for(i = 0; i < 256; i++)
            if(count[i] != 0)
                return -1;

        i = A.length() - 1;
        j = B.length() - 1;

        while(i >= 0)
        {
            // If there is a mismatch, then
            // keep incrementing result 'res'
            // until B[j] is not found in A[0..i]
            if(A.charAt(i) != B.charAt(j))
                res++;
            else
```

```
                j--;
            i--;
        }
        return res;
    }

    // Driver code
    public static void main(String[] args)
    {
        String A = "EACBD";
        String B = "EABCD";

        System.out.println("Minimum number of "
                    + "operations required is "
                                + minOps(A, B));
    }
}

// This code is contributed by Dipesh Jain
// (dipesh_jain)
```

From <https://www.geeksforgeeks.org/transform-one-string-to-another-using-minimum-number-of-given-operation/>