

**Submitted to:**

**Mam Memoona Malik**

**Submitted by:**

1. **Ali Hamza**

(FA19-BCS-013)

1. **Khadija Tuz Zehra**

(FA19-BCS-033)

1. **Mahrukh**

(FA19-BCS-034)

1. **Mohammad Haris Zia**

(FA19-BCS-037)

**Assignment 03**

Design and Analysis of Algorithms

**COMSATS University Islamabad**

Q1: a) Explain the longest common subsequence problem.

**Longest Common Subsequence:**

The longest common subsequence (LCS) is defined as the longest subsequence that is shared by all the supplied sequences if the subsequence's components do not have to be in consecutive locations in the original sequences.

**LCS Problem Statement:**

The challenge of identifying the longest subsequence common to all sequences in a set of sequences is known as the longest common subsequence (LCS) (often just two sequences).

**Explanation:**

Consider two strings, string 1 and string 2:

*String 1: a b c d e f g h i j*

*String 2: c d g i*

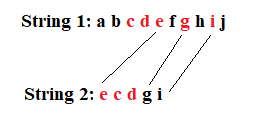
String 2 is a subsequence of String 1, even though they are not in consecutive locations but the order of the characters in string 2 matches with String 1.

**Example:**

*String 1: a b c d e f g h i j*

*String 2: c d g i*

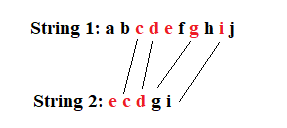
Solving, for the following strings:



Here, we cannot match “d and c” of String 1 with “d and c” of String 2, because it comes before e in String 1 and this is a restriction in the LCS problem that we can’t do matching if the character appears in this way.

So, the first sequence is:

**e g i**

****

Here, we cannot match “e” of String 1 with “e” of String 2 as we must complete matching without intersection.

And the second possible sequence is:

**c d g i**

Common Subsequence:

1. e g i (length =3)
2. c d g i (length =4)

Longest Common Subsequence:

**LCS is “c d g i” with length 4.**

b) Describe its time complexity with simple recursive solution.

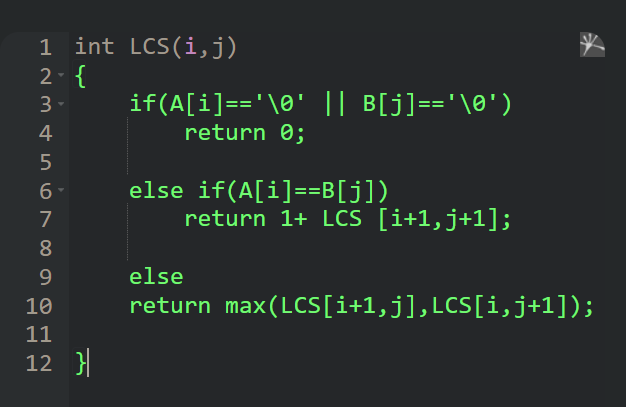
**Recursion:**

***Step1:*** Finding all the possible subsequences of string X takes 2^m times where m is the length of the string.

***Step2:*** To check if it is a common subsequence in both strings i.e., String X and String Y where length of String Y is n it takes O(n) times.

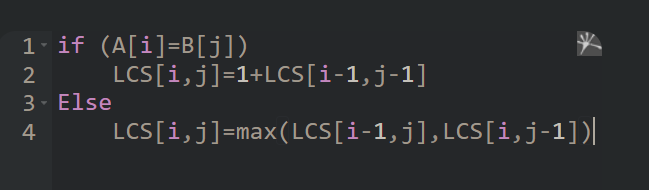
So, the total time complexity will be combined of both steps i.e., O(n2^m).

**Recursive Solution:**

****

c)Write an algorithm to find the longest common subsequence using dynamic programming approach

**Algorithm to find LCS using Dynamic Programming:**

****