**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Computer Engineering Department**

Program: B.Tech. Sem V

**Course: Design and Analysis of Algorithms**

**List of Experiments**

w.e.f. 1st Jul 2020

**Faculty:** Abhay Kolhe.

LAB Manual

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**Experiment No.07**

**A.1 Aim:**

Implementation of Dynamic Programming Technique Algorithm Design.

Write a program to implement Longest Common Subsequence (LCS) problem.

**A.2 Prerequisite:**

1. Concepts of Dynamic Programming Technique of algorithm design.

2. Knowledge of Matrix Handling.

3. Knowledge of Implementing Recursion.

**A.3 Outcome:**

**After successful completion of this experiment students will be able to**

1. Design & implement a solution using Dynamic Programming Technique.
2. Identify different problems that can be solved by using Dynamic Programming Technique.
3. Identify applications of LCS Problem.

**A.4 Theory:**

**A.4.1.**

**A subsequence of a** sequence/string *X* = < *x* , *x* , …, *x* > is a sequence obtained by deleting 0 or more elements from *X*.

**Example: “sudan” is a subsequence of “sesquipedalian”.**

**So is “equal”.**

**There are 2 subsequences of *X* .**

**A common subsequence *Z* of two sequences *X* and *Y* is a subsequence of both.**

**Example: “ua” is a common subsequence of “sudan” and “equal”.**

The **longest common subsequence problem** (LCS) is finding a longest sequence which is a subsequence of all sequences in a set of sequences (often just two). The problem is sometimes defined to be finding all longest common subsequences.

It should not be confused with the longest common substring problem (a substring is necessarily a contiguous part).

Input: *X* = < *x*1 , *x2* , **…**, *xm* >

*Y* = < y1 , y2 , **…**, yn >

Output: a *longest common subsequence* (LCS) of *X* and *Y*.

Example a) *X* = abcbdab *Y* = bdcaba

LCS1 = bcba LCS2 = bdab

b) *X* = enquiring *Y* = sesquipedalian

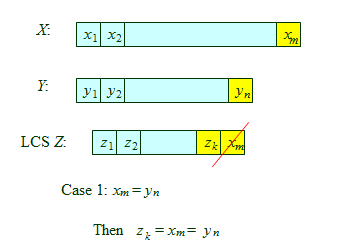
LCS = equiin

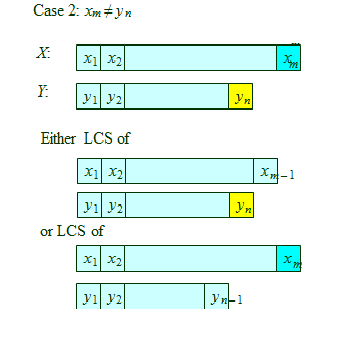
c) *X* = empty bottle *Y* = nematode knowledge

LCS = emt ole

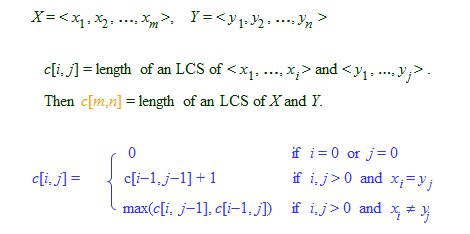
## Solution for two sequences

Optimal Substructure of LCS





A Recursive Formula for this



**A.5 Procedure/Algorithm:**

**A.5.1:**

**LCS Algorithm**

m ← length[X}

n ← length[Y]

for i ← 1 to m do  
 c[i, 0] ← 0

for j ← 1 to n do  
 c[0, j] ← 0

for i ← 1 to m do  
 for j ← 1 to n do  
 if xi = yj  
 c[i, j] ← c[i-1, j-1]+1  
 b[i, j] ← “D ”   
 else

if c[i-1, j] ≥ c[i, j-1]  
 c[i, j] ← c[i-1, j]  
 b[i, j] ← “U ↑”  
 else  
 c[i, j] ← c[i, j-1]  
 b[i, j] ← “L ←”

return c and b

**To find an LCS follow the arrows of matrix b, for each diagonal arrow ( ) there is a member of the LCS.**

***c*[*m*,*n*] = length of an LCS of *X* and *Y*.**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)***

|  |  |
| --- | --- |
| Roll No. | Name: |
| Class : | Batch : |
| Date of Experiment: | Date of Submission |
| Grade : | Time of Submission: |
| Date of Grading: |  |

**B.1 Software Code written by student:**

***(Paste your c/c++ code completed during the 2 hours of practical in the lab here)***

**B.2 Input and Output:**

***(Paste your program input and output in following format, If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)***

**Input Data:**

**Output Data:**

**B.3 Observations and learning:**

***(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)***

**B.4 Conclusion:**

*(****Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)***

**B.5 Question of Curiosity**

***(To be answered by student based on the practical performed and learning/observations)***

Q.1 Identify & discuss the real life applications LCS.

Q.2 Why do we need the matrix B/b in the above algorithm?

Q.3 Devise an algorithm to print the LCS in forward direction.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*