DAA:- IT-081 Pankhania Aanandi R.

Longest Common Subsequence.

Analysis of the problem:-

The longest common subsequence problem is finding the longest subsequence which exist in both the given strings.

Let $X = \langle x_1, x_2, x_3, \ldots, x_m \rangle$ & $Y = \langle y_1, y_2, y_3, \ldots, y_n \rangle$ be the sequences.

then we can say that Z is the common subsequence of X and Y, if Z is a subsequence of both X & Y.

[$Z = \langle z_1, z_2, \ldots, z_n \rangle$ for eg. let]

In this procedure, table C[m,n] is computed in either ~~RO~~ RMO or CMO

(Row major order)
(Column " ")

Design of the Algorithm :-

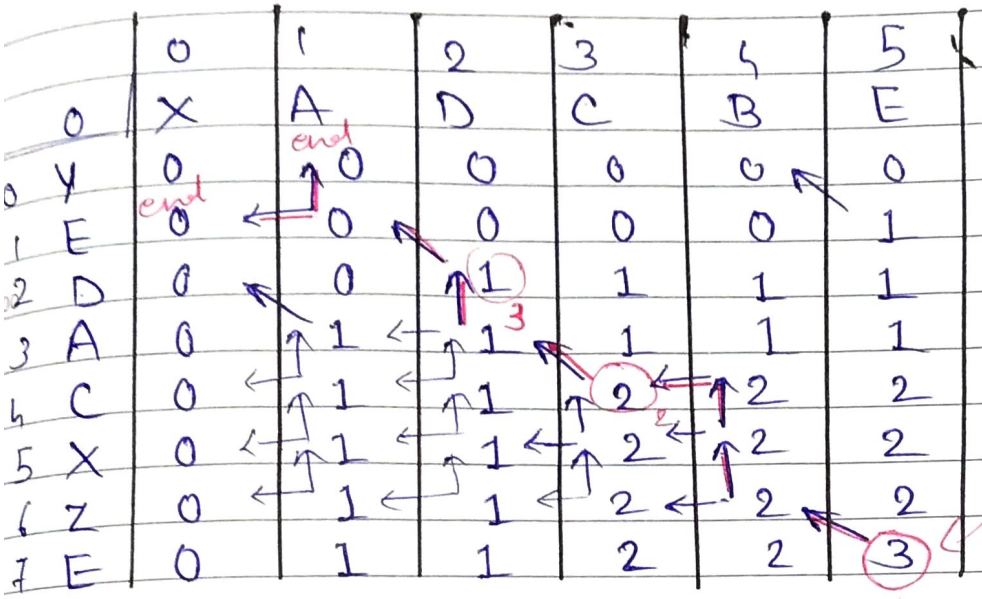
Algo: LCS-length-table-formulation(X, Y)

```
m := length(X)
n := length(Y)
for i = 1 to m do
    c[i, 0] := 0   // [first in every row = 0]
for j = 1 to n do
    c[0, j] := 0   // [first in every coln = 0]
for i = 1 to m do
    for j = 1 to n do
        if xi = yj
            c[i, j] = c
            // B[i, j] := 'D'
        else
            if c[i-1, j] ≥ c[i, j-1]
                c[i, j] = c[i-1, j] + 1
                // B[i, j] = 'U'
            else
                c[i, j] := c[i, j-1]
                // B[i, j] := 'L'
return C
```

example

X = A D C B E
Y = E D A C X Z E

→ Ans: DCE



Analysis:-

To populate the table, the outer for loop iterates m times and the inner for loop iterates n times.

Hence,

Time & Space complexity = $O(mn)$

{∵ m, n length of two strings}