Jawahar Education Societys Annasaheb Chudaman Patil College of Engineering, Kharghar, Navi Mumbai

NAME: PRIYUSH BHIMRAO KHOBRAGADE

PRN NO: 211112018

Roll No: 52

SUBJECT: Analysis of Algorithms Lab

Implement a C program for the Longest Common Subsequence

EXPERMINT: 08

PAGE NO.:
DATE.: / / 20

Experiment No-08

- **Aim:-** Write 'C' program for longest common subsequence using Dynamic Programming Approach.

- **Objective :** To implement and analyse complexity of longest common subsequence.

- **Hardware / Software Required :-** Turbo 'C'

- **Outcomes:-** Students will take or able to compute the complexity of longest common subsequence.

- **Problem Definition**

The two strings $X = x_1, x_2 \cdots x_m$ and $Y = y_1, y_2 \cdots y_n$ of length m and n are given. The objective function is to determine the longest string S that is a subsequence of both X & Y.

- **Theory:-**

Dynamic programming :-

Dynamic programming is an algorithm design method that can be used when the solution to the problem can be viewed as the result of a sequence of decision. It avoids recomputing solution that have already been computed. D.P uses "principle of optimality."

The principle of optimality state that "in an optimal sequence of decisions or choices, each subsequence must also be optimal."

Teachers Signature _____ 1

Implement a C program for the Longest Common Subsequence

PAGE NO.:
DATE.: / / 20

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

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

$$L[i,j] = \begin{cases} 0 \\ L(i-1, j-1)+1 \\ \max\{L(i-1,j), L(i,j-1)\} \text{ otherwise.} \end{cases}$$

Since this problem has an optimal substructure property, it can be solved by dynamic programming.

The relational for this recurrence is that, if the last character of two sequences are equal, they must be part of the LCS. A longer LCS can never be obtained by matching x_m to y_i when i <n & vice versa. To find all the longest common-subsequence, the LCS should be denoted by a set of sequence, and 'max should select both solution if they are equally long. If choosing X_i and Y_j should give an equally long result (both resulting subsequences should be shown. This is returned as a set by this function. Notice that this function is not polynominal, as it might branch in almost every step if the strings are similar.

Teachers Signature _____

Implement a C program for the Longest Common Subsequence

PAGE NO.:
DATE.: / / 20

- Procedure / Algorithm :-

```
for i ← 1 to m do
    for j ← 1 to n do
      if x_i = y_j
      L[i,j] ← L[i-1, j-1] + 1
      b[i,j) ← " "
Else
      if L[i-1, j] ≥ L[i, j-1]
    L[i,j] ← L[i-1, j)
    b[i,j] ← " "
else
    L[i,j] ← L[i, j-1]
    b[i,j] = " "
```

- Analysis :-

$$T(n) = \sum_{i=1}^{m} 1 \cdot \sum_{j=1}^{n} 1.$$

$$= mn = O(mn)$$

- Conclusion :-

Thus, it is observed that the complexity of LCS problem is $O(mn)$.

3

Teachers Signature _____

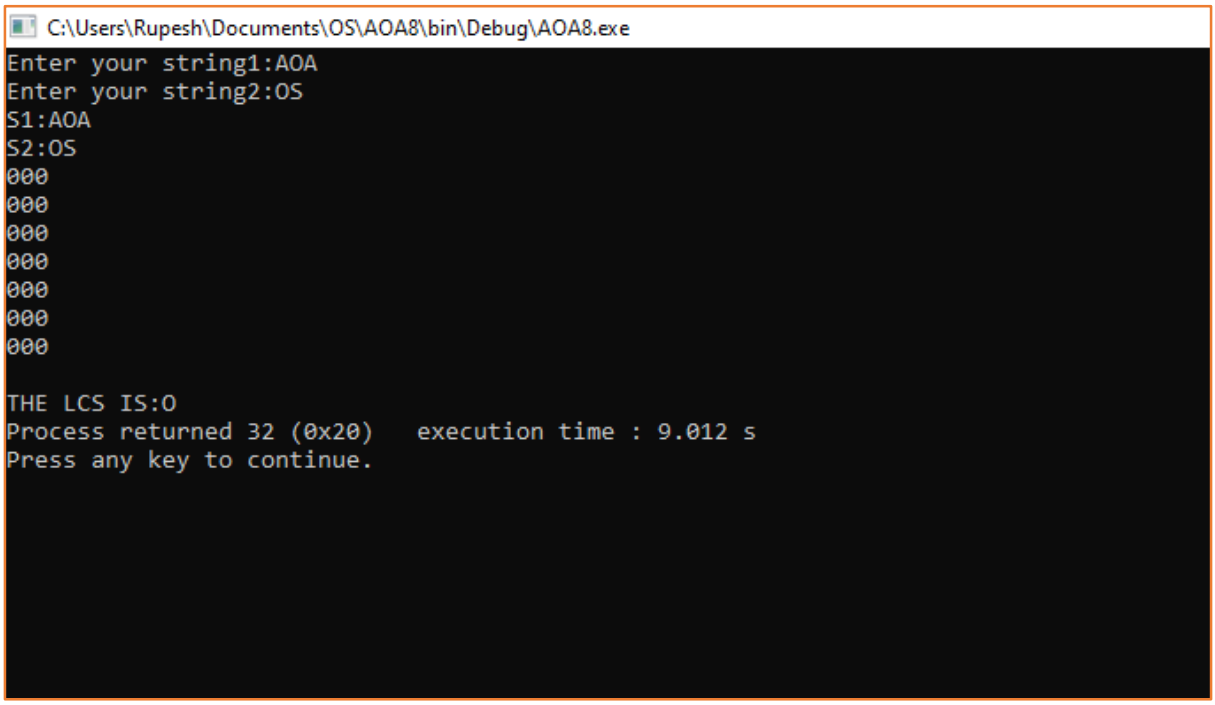
Implement a C program for the Longest Common Subsequence

Input:

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
char b[20][20],x[20],y[20];
int c[20][20];
int m,n;
void lcss(char x[],char y[])
{
int i,j;
m=strlen(x);
n=strlen(y);
for(i=0;i<=m;i++)
{
c[i][0]=0;
}

for(i=0;i<=n;i++)
{
c[0][i]=0;
}
printf("\n");
for(i=0;i<=m;i++)
{
for(j=0;j<=n;j++)
{
printf("%d",c[i][j]);
}
printf("\n");
}

for(i=1;i<=m;i++)
{
for(j=1;j<=n;j++)
{
for(j=0;j<=n;j++)
printf("%d",c[i][j]);
printf("\n");
for(j=1;j<=n;j++)

if(x[i-1]==y[j-1])
{
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';
}
}
}
}

void display(int i,int j)
{
if(i==0||j==0)
return;
if(b[i][j]=='d')
{
display(i-1,j-1);
printf("%c",x[i-1]);
}
else if(b[i][j]=='u')
display(i-1,j);
else
display(i,j-1);
}

void main()
{
printf("Enter your string1:");
scanf("%s",x);
printf("Enter your string2:");
scanf("%s",y);
printf("S1:%s",x);
printf("\nS2:%s",y);
lcss(x,y);
printf("\nTHE LCS IS:");
display(m,n);
getch();
}

```

Implement a C program for the Longest Common Subsequence

Output:



Conclusion: Thus it is observed that the Complexity **of Longest Common Subsequence LSC** Problem is O(mn).