Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058, India
(Autonomous College Affiliated to University of Mumbai)

Experiment No.	**5**
Aim	**Dyanamic Programming-Longest Common Subsequence**
Name	**Pooja Gajendra Patil**
UID No.	**2022301011**
Class & Division	**Computer Engineering-A (A3)**
Date of Performance.	**20-03-23**
Date of Submission.	**27-03-23**

Theory:

Dynamic programming is a technique used to solve optimization problems by breaking them down into smaller subproblems and solving them in a recursive manner. One classic example of dynamic programming is the longest common subsequence problem.

The longest common subsequence (LCS) is defined as the longest subsequence that is common to all the given sequences, provided that the elements of the subsequence are not required to occupy consecutive positions within the original sequences.

If *S1* and *S2* are the two given sequences then, *Z* is the common subsequence of *S1* and *S2* if *Z* is a subsequence of both *S1* and *S2*. Furthermore, *Z* must be a **strictly increasing sequence** of the indices of both *S1* and *S2*.

In a strictly increasing sequence, the indices of the elements chosen from the original sequences must be in ascending order in *Z*.

Let us understand LCS with an example.

If

```
S1 = {B, C, D, A, A, C, D}
S2 = {A, C, D, B, A, C}
```

Then, common subsequences are `{B, C}`, `{C, D, A, C}`, `{D, A, C}`, `{A, A, C}`, `{A, C}`, `{C, D}`, `...`

Among these subsequences, `{C, D, A, C}` is the longest common subsequence.

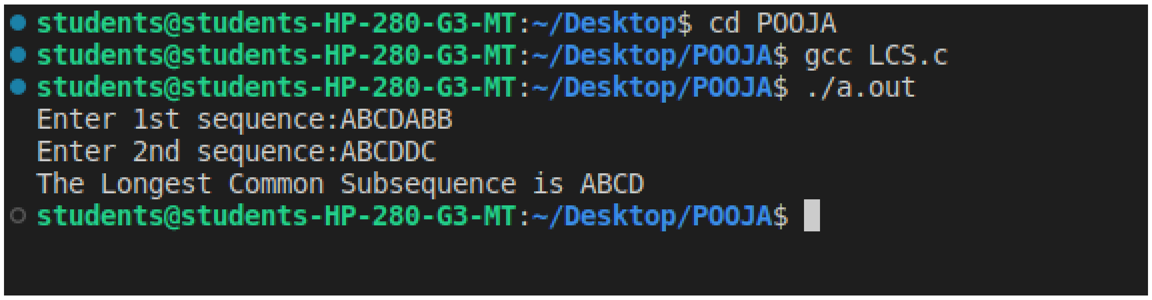
Here are the steps to find the LCS of two sequences:

1. Create a table of size (m+1)x(n+1) where m and n are the lengths of the two sequences. Initialize all the cells to zero.
2. Traverse through each cell of the table in row-major order. For each cell (i,j), if the ith character of sequence 1 matches the jth character of sequence 2, set the value of the cell (i,j) to the value of cell (i-1,j-1) + 1. Otherwise, set the value of the cell (i,j) to the maximum of the values of cell (i-1,j) and cell (i,j-1).
3. The value in the bottom-right corner of the table is the length of the LCS.
4. To reconstruct the LCS, start at the bottom-right corner of the table and work backwards. If the value of cell (i,j) is equal to the value of cell (i-1,j-1) + 1, then the ith character of sequence 1 and the jth character of sequence 2 are part of the LCS. Move diagonally up to the cell (i-1,j-1) and repeat the process. If the value of cell (i,j) is equal to the value of cell (i-1,j), move up to the cell (i-1,j) and repeat the process. If the value of cell (i,j) is equal to the value of cell (i,j-1), move left to the cell (i,j-1) and repeat the process.
5. Continue the process until the top-left corner of the table is reached. The characters visited in reverse order will form the LCS.

Program:

```
#include<stdio.h>
#include<string.h>
int i,j,m,n,c[20][20];
char x[20],y[20],b[20][20];

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

void lcs()
{
   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;

//c, u and l is cross, upward and downward respectively
    for(i=1;i<=m;i++)
    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]='c';
            }
        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';
        }
        }
}
int main()
{
    printf("Enter 1st sequence:");
    scanf("%s",x);
    printf("Enter 2nd sequence:");
    scanf("%s",y);
    printf("The Longest Common Subsequence is ");

    lcs();
    print(m,n);
    printf("\n");

return 0;
}
```

Output:



```
● students@students-HP-280-G3-MT:~/Desktop$ cd POOJA
● students@students-HP-280-G3-MT:~/Desktop/POOJA$ gcc LCS.c
● students@students-HP-280-G3-MT:~/Desktop/POOJA$ ./a.out
  Enter 1st sequence:ABCDABB
  Enter 2nd sequence:ABCDDC
  The Longest Common Subsequence is ABCD
○ students@students-HP-280-G3-MT:~/Desktop/POOJA$ 
```

Conclusion: Thus we have studied and implemented the longest common subsequence.