Aayush Shah
19BCE245
11 November 2021

Design and Analysis of Algorithms

Practical 8

• **Code :**

```
/*
Aayush Shah
19BCE245
DAA Practical 8 | Longest Common SubSequence
*/

/* Dynamic Programming C implementation of LCS problem */
#include <stdio.h>
#include <string.h>

//#define MIN(a,b) (((a)<(b))?(a):(b))
#define MAX(a,b) (((a)>(b))?(a):(b))


/* Returns length of LCS for X[0..m-1], Y[0..n-1] */
int lcs( char *X, char *Y, int m, int n )
{
     int L[m+1][n+1];
     int i, j;

     // Generating Matrix
     for (i=0; i<=m; i++)
     {
          for (j=0; j<=n; j++)
          {
               if (i == 0 || j == 0)
                    L[i][j] = 0;

               else if (X[i-1] == Y[j-1])
```

```
                L[i][j] = L[i-1][j-1] + 1;

            else
                L[i][j] = MAX(L[i-1][j], L[i][j-1]);
        }
    }

    // Getting longest common subsequence
    int index = L[m][n];

    char lcs[index+1];
    lcs[index] = '\0';

    int row = m, col = n;
    while (row > 0 && col > 0)
    {

        if (X[row-1] == Y[col-1])
        {
            lcs[index-1] = X[row-1];
            row--; col--; index--;
        }

        else if (L[row-1][col] > L[row][col-1])
            row--;
        else
            col--;
    }
    printf("\t> Longest Common Subsequence : %s.\n",lcs);

    return L[m][n];
}

int main()
{
    char str1[100], str2[100];
    printf("Enter first string : ");
    scanf("%s",str1);
    printf("Enter second string : ");
    scanf("%s",str2);

    int m = strlen(str1);
    int n = strlen(str2);
```

```
    printf("\t> Length of LCS is %d.", lcs( str1, str2,
m, n ) );

    return 0;
}
```

• **Output :**