|  |  |
| --- | --- |
| **Experiment No.** | **5** |
| **Aim** | **To implement Longest Common Subsequence Algorithm using Dynamic Programming .** |
| **Name** | **Atharva Gite** |
| **UID No.** | **2021300038** |
| **Class & Division** | **COMPS-A(C)** |

**Theory:**

The Longest Common Subsequence (LCS) problem is a classic problem in computer science that involves finding the longest subsequence that is common to two or more sequences. A subsequence is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements.

The LCS problem is important in applications such as text comparison, DNA sequencing, and bioinformatics. It can be solved using dynamic programming, which involves breaking the problem down into smaller subproblems and solving them recursively.

Pseudo-Code:

1. Start
2. Take two input sequences, X and Y
3. Find the lengths of sequences X and Y and store them in variables m and n
4. Create an empty matrix L with dimensions (m+1) x (n+1)
5. For each element in matrix L, starting from the first row and column, do the following:

a. If either i or j is 0, set L[i][j] to 0

b. Otherwise, if the i-th element of X is equal to the j-th element of Y, set L[i][j] to the value of the diagonal element plus 1 (i.e., L[i-1][j-1]+1) c. Otherwise, set L[i][j] to the maximum of the element above it and the element to its left (i.e., max(L[i-1][j], L[i][j-1]))

1. Return the value in the bottom-right corner of matrix L, which represents the length of the longest common subsequence
2. End

**CODE:**

#include <stdio.h>

#include <string.h>

// function to find the longest common subsequence of two strings

void lcs(char \*X, char \*Y, int m, int n) {

int L[m+1][n+1]; // 2D array to store the lengths of LCS

// build the L[m+1][n+1] matrix bottom-up

for (int i = 0; i <= m; i++) {

for (int 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] = (L[i-1][j] > L[i][j-1]) ? L[i-1][j] : L[i][j-1];

}

}

}

// print the longest common subsequence

int index = L[m][n];

char lcs[index+1];

lcs[index] = '\0';

int i = m, j = n;

while (i > 0 && j > 0) {

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

lcs[index-1] = X[i-1];

i--;

j--;

index--;

} else if (L[i-1][j] > L[i][j-1]) {

i--;

} else {

j--;

}

}

printf("The longest common subsequence is: %s\n", lcs);

}

int main() {

char X[100], Y[100];

printf("Enter the first string: ");

scanf("%s", X);

printf("Enter the second string: ");

scanf("%s", Y);

int m = strlen(X);

int n = strlen(Y);

lcs(X, Y, m, n);

return 0;

}

**Result:**



