# California State University, Fresno

# DEPARTMENT OF COMPUTER SCIENCE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class: | **Algorithms & Data Structures** | | | Semester: | **Fall 2021** |
|  | | | | | |
| Points |  | Document author: | **Saishnu Ramesh Kumar** | | |
|  | Author’s email: | **saishnu\_rk@mail.fresnostate.edu** | | |
| Laboratory number: | **Section 1, 11am to 12:50pm** | | |
|  | | | | | |

**1. Statement of Objectives**

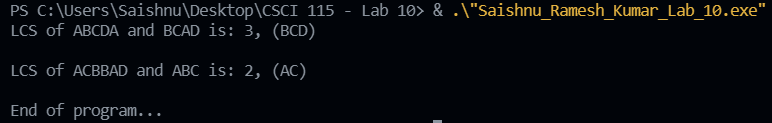
This assignment is a continuation of Dynamic Programming from the previous lab assignment. In this exercise, we were assigned to make a program for the LCS (Longest Common Subsequence). The LCS would basically find the longest common subsequence of two sequence of items. The time complexity of LCS in Dynamic Programming is O(n \*m) where n and m are the lengths of the string.

**2. Experimental Procedure**

For this assignment, I created a LCS function that takes two string parameters and two integer parameters. Within the LCS function, I created a 2-dimensional array and also had created a nested for loop. Within the for loop, there is the conditional statements that were mentioned during the lab session that we were asked to include as well for the c[i, j] matrix. The while loop would get the output for the longest LCS as well as getting the length of the final output. In the main function I defined four strings for two different outcomes, x1, y1, x2, and y2, and created two tries which is basically two different outcomes and the LCS function is then called to output the length and the longest LCS.

**3. Analysis**

Screenshot of Terminal Output:



**4. Encountered Problems**

This concept is still new to me, and I am learning more as I code. I encountered syntax errors and some logical errors in my code, and I was able to fix them.

**5. Conclusions**

From this lab assignment, I have learnt how to code the Longest Common Subsequence example. I believe that this example was much easier to get a understanding of as compared to the Matrix Multiplication Order code that we were assigned the week before. The way that LCS works is interesting as it takes the longest common list of items from two sequences and puts them into a much more simplified version of one list.

**6. References**

Slides provided by TA

<https://cppsecrets.com/users/101611099712197110107114971065149535048484864103109971051084699111109/C00-Longest-Common-Subsequence-using-DP.php>

<https://www.tutorialspoint.com/cplusplus-program-for-longest-common-subsequence>

<https://www.geeksforgeeks.org/cpp-program-for-longest-common-subsequence/>

<https://www.programiz.com/dsa/longest-common-subsequence>