

Lab Assignment 05

Task 1: Implement an algorithm to perform the BFS traversal of a graph. Display the order of traversed vertices and the final output of the BFS traversal.

Task 2: Implement the Topological Sort algorithm. Mention the finishing time of each vertices. List the possible topologically sorted ordering of all of graph's vertices.

Task 3: Consider the problem of finding the **Longest Common Subsequence** between given two sequences of strings X of length m and Y of length n as $X = \langle x_1, x_2, \dots, x_m \rangle$ and $Y = \langle y_1, y_2, \dots, y_n \rangle$, where $x_i, 1 \leq i \leq m$ and $y_j, 1 \leq j \leq n$ are members of a finite set of symbols.

The above problem can be solved using the following approaches:

1. **Apply Brute Force (*Exhaustive*) approach** ([ie. for every subsequence of X, check if it is a subsequence of Y), to find the longest common subsequence between two sequences X and Y. Print the longest common subsequence and its length.
2. **Dynamic Programming approach:** Use dynamic programming approach to find the the longest common subsequence between X and Y sequences. Display the completely filled up C table storing length of LCS and B table that records the information to generate the optimal solution. You display the obtained final LCS.

Comparatively analyze the performance of above two approaches by computing execution time taken by each approach for the same set of input data. Store the time taken along with the value of input size used into a file for plotting a graph. Create program profile and analyze the running time. Do the performance evaluation by using gprof. Write your program using modules and multi file programming approach i.e. your program file divided into multiple files and programs into modules.