Design and Analysis of Algorithms Practice
LAB 8
Date: $15^{th}$ October 2024

**General instructions:**

1. Students have to write the pseudo code first in their notebooks and implement it after that. Students can use either C / C++.

2. The point of contact (Member 1 as submitted in Gform) from the group has to submit all the programs. You may ask the TA, if you forgot the point of contact (Member 1).

3. Submit all the programs as a single Zip file in Google Class Room (GCR).

4. Pseudo code, Demonstration and Viva will be evaluated by the TA for 10 marks each and a total of 30. Pseudo code and Viva will be evaluated in the lab itself.

5. If the students wish to submit the programs later, then they can do it with in 2 days (i.e., if the lab is on Tuesday, then programs need to be submitted by Thursday 11:59 PM by point of contact (Member 1).). This evaluation will be considered for Demonstration 10 marks.

**All about the Dynamic Programming Paradigm**

Q1) **0/1 Knapsack Problem:** Given a set of $n$ objects $S = \{X_1, X_2, ..., X_n\}$ associated with their profits $\{P_1, P_2, ..., P_n\}$, weights $\{W_1, W_2, ..., W_n\}$ and Capacity constraint (C) of the Knapsack. The following equation is a formulation of 0/1 knapsack problems.

$$\text{Maximize } \sum_{i=1}^{n} P_i X_i$$
$$\text{Such that } \sum_{i=1}^{n} W_i X_i \leq C,$$
$$X_i \in \{0, 1\}$$

Design and implement the dynamic programming strategy and evaluate its associated time complexity in terms of Asymptotic Notations (Big O / Theta).
Attached problem instances are described in the following data format in .kp file.
n
C
$P_1, W_1$
$P_2, W_2$
..
$P_n, W_n$

Q2) **Matrix Chain Multiplication Problem:** Given $n$ matrices $A_1, A_2, ..., A_n$ and $(P_0, P_1, ..., P_n) = n+1$ integer values. To produce the output: optimal order of multiplications (i.e., Optimal Parenthesization), design and implement the dynamic programming strategy and evaluate its associated time complexity in terms of Asymptotic Notations (Big O / Theta).

Q3) **Longest Common Subsequence Problem:** Given two strings $X, Y$ which are of length $M, N$ respectively. To find out a longest common subsequence, design and implement the dynamic programming strategy and evaluate its associated time complexity in terms of Asymptotic Notations (Big O / Theta).

**Note:** Students who have finished the above programs in less time can explore other problems using dynamic programming strategy.