CSP214 TASKs #4

Task 4.1

Consider the ***matrix-chain multiplication*** problem which can be specified as follows: Given a chain <A_1,A_2,….,A_n> of n matrices , where i=1,2,3..n, matrix A_i has dimension of $p_{i-1} * p_i$, fully parenthesize the product A_1,A_2,….,A_n in a way that minimizes the number of scalar multiplications.

The above problem can be solved using the following approaches:

1. ***Exhaustive(Direct) Method:***
   Determine minimum number of multiplication required by computing number of scalar multiplication needs from all possible combination of matrix sequences (as brute force).
   Display all sequences along with the number of multiplication needed and the final one which have minimum scalar multiplication cost.
2. ***Dynamic Programming Method:***
   Find an optimal parenthesization of a matrix-chain product and the least number of scalar multiplication required to compute the matrix chain product using dynamic programming approach.. Display the completely filled up *M* table storing cost of scalar multiplications and *S* table that records value of *k* such that the optimal parenthesization of A_i A_{i+1}…A_j splits the product between A_k and A_{k+1}.

   Display the matrix name and its dimensions which you will use to test your program.

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.

Task 4.2

Design and implement an algorithm of finding the **Longest Common Subsequence between given THREE sequences** of strings P of length l, Q of length m and R of length n as P=<p_1,p_2…p_l >, Q=<q_1,q_2…q_m >, and R= <r_1,r_2,…r_n > , where p_i, $1\leq i \leq l$, q_j , $1\leq j \leq m$ and r_j , $1\leq k \leq n$ are members of a finite set of symbols.

Solve the above problem using Dynamic programming approach. Your devised algorithm must be specified as per the four basic steps of dynamic programming i.e. i) Characterize the LCS problem ii) Recursive equation of solution iii) Recursive algorithm based on the recursive solution to determine length of an LCS of three sequences iv) print the LCS of three sequences.

You can make any suitable assumption (if missing) wherever necessary with valid justification.
Perform the performance analysis of your code.