Bennett University Greater Noida
Department of CSE

Subject Lab: Algorithms & Complexity **Lab Duration: 10:40-12:35**
Lab Code: ECSE202L **Max Marks: 10**

Submission Guidelines:

1. The purpose of the course is to learn and analyse the complexity of multiple algorithms.
2. You are supposed to do this assignment on your own. While you may discuss the problem with other students, you are not allowed to copy any part of the code from other students or copy from any other source. Any form of **plagiarism** is not acceptable. If there is a substantial overlap between codes, students will get a reduction in their respective course grade.
3. The assignment should be **shown to the lab instructor** in the lab session and **must be submitted** on LMS by **given date**.

It should also carry the following statement:

"***I have done this assignment on my own. I have not copied any code from another student or any online source. I understand if my code is found similar to somebody else's code, my case can be sent to the Disciplinary committee of the institute for appropriate action.***"

Lab Assignment 7

Q1. Write the code for Fibonacci sequence in the following two ways:

a) Recursive version.
b) Dynamic programming.

For each of the two sub-problems, you have to compute the execution time and return the number of computations.

The Fibonacci series is given as:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ……..

In mathematical terms, the sequence F_n of Fibonacci numbers is defined by the following recurrence relation:

$$F_n = F_{n-1} + F_{n-2}$$

with seed values

$$F_0 = 0 \text{ and } F_1 = 1.$$

Bennett University Greater Noida
Department of CSE

Subject Lab: Algorithms & Complexity **Lab Duration: 10:40-12:35**
Lab Code: ECSE202L **Max Marks: 10**

In your program, use large values of "n".

```
Input: n = 2
Output: 1,1
Execution Time: xxx ms %%% Correct this.
Number of Computations: %%% Correct this.


Input: n = 9
Output: Write the list here.
Execution Time: yyyyy ms %%% Correct this.
Number of Computations: %%% Correct this.
```

Q2a. Given a sequence of matrices, write a program to find the most efficient way to multiply matrices together. Note: You need not perform the multiplications, the goal is to merely decide upon the order of multiplication.

Input: p[ ] = {40, 20, 30, 10, 30}

Output: 26000

There are 4 matrices of dimensions 40x20, 20x30, 30x10 and 10x30.

Let the input 4 matrices be A, B, C and D. The minimum number of multiplications are obtained by putting parenthesis in following way

(A(BC))D --> 20*30*10 + 40*20*10 + 40*10*30

Input: p[] = {10, 20, 30, 40, 30}

Output: 30000

There are 4 matrices of dimensions 10x20, 20x30, 30x40 and 40x30. Let the input 4 matrices be A, B, C and D. The minimum number of multiplications are obtained by putting parenthesis in following way

((AB)C)D --> 10*20*30 + 10*30*40 + 10*40*30

Bennett University Greater Noida
Department of CSE

Subject Lab: Algorithms & Complexity **Lab Duration: 10:40-12:35**
Lab Code: ECSE202L **Max Marks: 10**

Input:
The first line of each test case contains an integer **N**, denoting the number of elements in the array. Then next line contains **N** space separated integers denoting the values of the element in the array.

Output:

For each test case the print the minimum number of operations needed to multiply the chain with the proper place of the parenthesis.

Example 1:
Input:
5 # no. of inputs
1 2 3 4 5 # Inputs
Output:
38 1(2(3(45)))

Example 2:
Input:

3 # No. of inputs
3 3 3 # inputs
Output:
27 ((12)3)

Q2b Write an 'LCS' function that takes two sequences and finds the length of the longest common subsequence.

Function prototype:
"int lcs(char *X, char *Y, int m, int n)"

A subsequence is a sequence that appears in the same relative order, but not necessarily contiguous. For example, “abc”, “abg”, “bdf”, “aeg”, ”acefg”, .. etc are subsequences of “abcdefg”.

Example:

Input:
ABCDGH

AEDFHR

Bennett University Greater Noida
Department of CSE

Subject Lab: Algorithms & Complexity **Lab Duration: 10:40-12:35**
Lab Code: ECSE202L **Max Marks: 10**

Final Output:
3 ADH

Interpretation :
common 'ADH' sub-sequence is of length 3. (A, D, H)