

Design Analysis and Algorithm Lab

M.C.A.-Semester-I



Session: Winter 2024 (2024-2025)

Ramdeobaba University, Nagpur

**Department of Computer Science and
Application**

**MCA(Artificial Intelligence and Machine
Learning)**

Name of Program: Implementation of Chain Matrix Multiplication using dynamic programming technique..

Source Code:

```
import java.util.*;

class ChainMatrix {

    static char name;

    static void printParenthesis(int i, int j, int[][] bracket) {
        // Base case: If only one matrix is left in the current segment
        if (i == j) {
            System.out.print(name++);
            return;
        }

        System.out.print('(');

        // Recursively print the left side of the chain
        printParenthesis(i, bracket[i][j], bracket);

        // Recursively print the right side of the chain
        printParenthesis(bracket[i][j] + 1, j, bracket);

        System.out.print(')');
    }

    // Matrix Ai has dimension p[i-1] x p[i] for i = 1..n
    static void matrixChainOrder(int[] p, int n) {
```

```

int[][] m = new int[n][n];
int[][] bracket = new int[n][n];

// Initialize the diagonal elements to 0 because cost is 0 when multiplying one matrix
for (int i = 1; i < n; i++) {
    m[i][i] = 0;
}

// Calculate the minimum cost for each subproblem of length L
for (int L = 2; L < n; L++) {
    for (int i = 1; i < n - L + 1; i++) {
        int j = i + L - 1;
        m[i][j] = Integer.MAX_VALUE;

        // Try all possible split points and compute the minimum cost
        for (int k = i; k <= j - 1; k++) {

            int q = m[i][k] + m[k + 1][j] + p[i - 1] * p[k] * p[j];
            if (q < m[i][j]) {
                m[i][j] = q;
                bracket[i][j] = k;
            }
        }
    }
}

// The first matrix is printed as 'A', next as 'B', and so on
name = 'A';

System.out.print("Optimal Parenthesization is: ");
printParenthesis(1, n - 1, bracket);

```

```

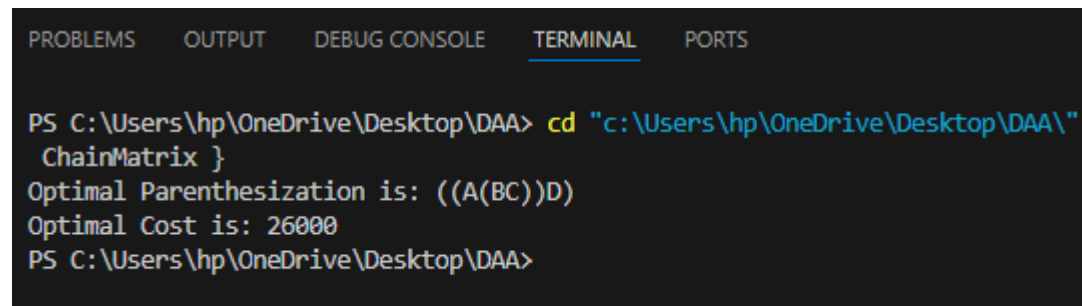
        System.out.print("\nOptimal Cost is: " + m[1][n - 1]);
    }

    public static void main(String[] args) {
        // Array representing matrix dimensions: { 40, 20, 30, 10, 30 }
        // This means 4 matrices with dimensions: A1(40x20), A2(20x30), A3(30x10),
        A4(10x30)
        int[] arr = { 40, 20, 30, 10, 30 };
        int n = arr.length;

        matrixChainOrder(arr, n);
    }
}

```

Output:



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\hp\OneDrive\Desktop\DAA> cd "c:\Users\hp\OneDrive\Desktop\DAA\"
ChainMatrix }
Optimal Parenthesization is: ((A(BC))D)
Optimal Cost is: 26000
PS C:\Users\hp\OneDrive\Desktop\DAA>

```

Submitted By:

Ritu Choudhary

Roll no:51

MCA(AI/ML) – Section B