Name: Ayush Patel Enrolment: 22162171038 Class B Batch 55

Institute of Computer Technology
B. Tech Computer Science and Engineering

Sub: Algorithm Analysis and Design

Practical 6

Given a sequence of matrices, we want to find the most efficient way to multiply these matrices together to obtain the minimum number of multiplications. The problem is not actually to perform the multiplication of the matrices but to obtain the minimum number of multiplications.

We have many options because matrix multiplication is an associative operation, meaning that the order in which we multiply does not matter. The optimal order depends only on the dimensions of the matrices.

The brute-force algorithm is to consider all possible orders and take the minimum. This is a very inefficient method.

Implement the minimum multiplication algorithm using dynamic programming and determine where to place parentheses to minimize the number of multiplications.

Find an optimal parenthesization of a matrix chain product whose sequence of dimensions are (5, 10, 3, 12, 5, 50, 6).

Code:

App.py

```
from flask import Flask, render_template, request

app = Flask(__name__)

def compute_chain_order(p_dims):
    matrix_count = len(p_dims) - 1
    cost = [[0 for _ in range(matrix_count)] for _ in range(matrix_count)]
          split = [[0 for _ in range(matrix_count)] for _ in
range(matrix_count)]

    for chain_len in range(2, matrix_count + 1):
```

Name: Ayush Patel Enrolment: 22162171038 Class B Batch 55

```
        for start in range(matrix_count - chain_len + 1):
            end = start + chain_len - 1
            cost[start][end] = float('inf')
            for mid in range(start, end):
                    current_cost = cost[start][mid] + cost[mid + 1][end] +
p_dims[start] * p_dims[mid + 1] * p_dims[end + 1]
                if current_cost < cost[start][end]:
                    cost[start][end] = current_cost
                    split[start][end] = mid

    return cost, split

def build_optimal_sequence(split_matrix, start, end):
    if start == end:
        return f"M{start + 1}"
    else:
        middle = split_matrix[start][end]
        left_part = build_optimal_sequence(split_matrix, start, middle)
        right_part = build_optimal_sequence(split_matrix, middle + 1, end)
        return f"({left_part} x {right_part})"

@app.route('/', methods=['GET', 'POST'])
def homepage():
    total_matrices = None
    total_operations = None
    optimal_sequence = None
    cost_matrix = None

    if request.method == 'POST':
                                        matrix_dims     =     list(map(int,
request.form['matrix_dims'].split(',')))
        total_matrices = len(matrix_dims) - 1
        cost_matrix, split_matrix = compute_chain_order(matrix_dims)
              optimal_sequence  =  build_optimal_sequence(split_matrix,  0,
total_matrices - 1)
        total_operations = cost_matrix[0][-1]

        # Mark non-diagonal zeros as 'N/A'
         cost_matrix = [['N/A' if cell == 0 else cell for cell in row] for
row in cost_matrix]
```

```
    return render_template('index.html',
                           total_operations=total_operations,
                           optimal_sequence=optimal_sequence,
                           cost_matrix=cost_matrix,
                           total_matrices=total_matrices)

if __name__ == '__main__':
    app.run(debug=True)
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Matrix Multiplication Optimization</title>
    <style>
        body {
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
            margin: 20px;
            background-color: #f9f9f9;
        }
        table {
            width: 100%;
            border-collapse: collapse;
            margin-top: 20px;
        }
        table, th, td {
            border: 1px solid #bdbdbd;
        }
        th, td {
            padding: 10px;
            text-align: center;
        }
        th {
            background-color: #e0e0e0;
```

Name: Ayush Patel Enrolment: 22162171038 Class B Batch 55

```
        }
        h1 {
            color: #333;
        }
        .matrix-info {
            font-weight: bold;
            color: #444;
        }
        input[type="text"] {
            padding: 8px;
            width: 300px;
            margin: 10px 0;
        }
        button {
            padding: 10px 15px;
            background-color: #4CAF50;
            color: white;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }
        button:hover {
            background-color: #45a049;
        }
    </style>
</head>
<body>
    <h1>Matrix Multiplication Optimization</h1>
    <form method="POST">
                    <label   for="matrix_dims">Enter   matrix   dimensions
(comma-separated):</label><br>
        <input type="text" id="matrix_dims" name="matrix_dims" required>
        <button type="submit">Compute</button>
    </form>

    {% if total_operations is not none %}
        <h2>Computation Results</h2>
        {% if total_matrices is not none %}
                <p  class="matrix-info">Total  matrices:  {{  total_matrices
}}</p>
```

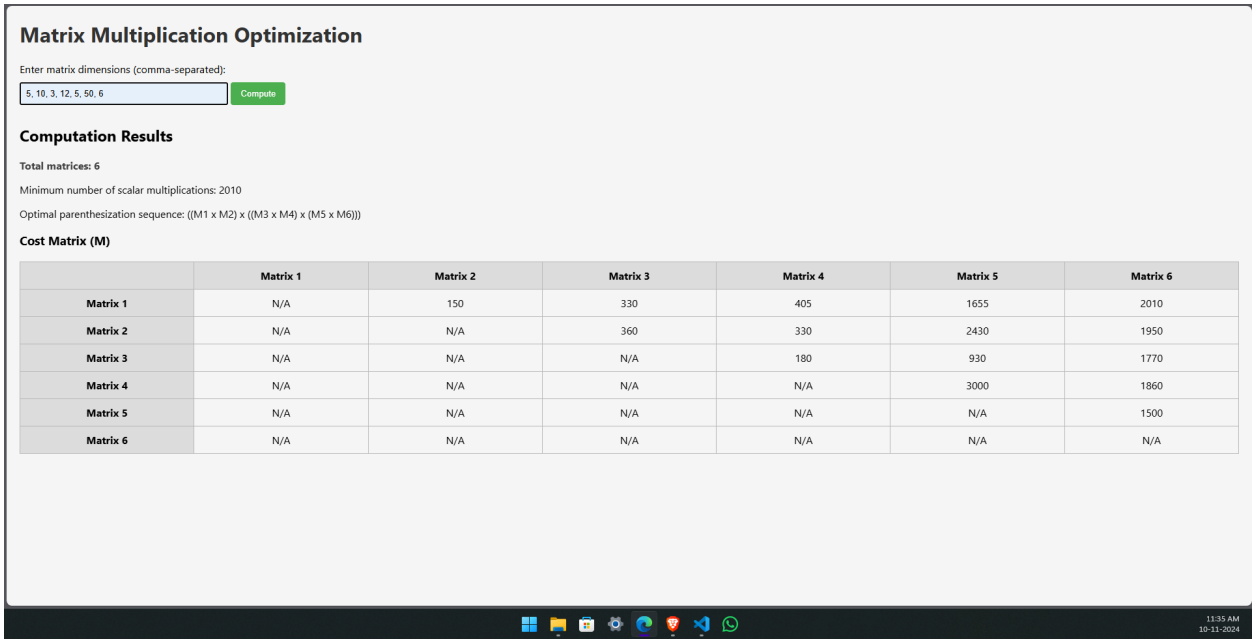
Name: Ayush Patel Enrolment: 22162171038 Class B Batch 55

```
        {% endif %}
          <p>Minimum number of scalar multiplications: {{ total_operations
}}</p>
        <p>Optimal parenthesization sequence: {{ optimal_sequence }}</p>

        {% if cost_matrix is not none %}
            <h3>Cost Matrix (M)</h3>
            <table>
                <tr>
                    <th></th>
                    {% for i in range(total_matrices) %}
                        <th>Matrix {{ i + 1 }}</th>
                    {% endfor %}
                </tr>
                {% for i in range(total_matrices) %}
                    <tr>
                        <th>Matrix {{ i + 1 }}</th>
                        {% for j in range(total_matrices) %}
                            <td>{{ cost_matrix[i][j] }}</td>
                        {% endfor %}
                    </tr>
                {% endfor %}
            </table>
        {% endif %}
    {% endif %}
</body>
</html>
```

Name: Ayush Patel Enrolment: 22162171038 Class B Batch 55

Output:



	Matrix 1	Matrix 2	Matrix 3	Matrix 4	Matrix 5	Matrix 6
Matrix 1	N/A	150	330	405	1655	2010
Matrix 2	N/A	N/A	360	330	2430	1950
Matrix 3	N/A	N/A	N/A	180	930	1770
Matrix 4	N/A	N/A	N/A	N/A	3000	1860
Matrix 5	N/A	N/A	N/A	N/A	N/A	1500
Matrix 6	N/A	N/A	N/A	N/A	N/A	N/A