Institute of Computer Technology

B. Tech Computer Science and Engineering

Practical 06

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).

App.py

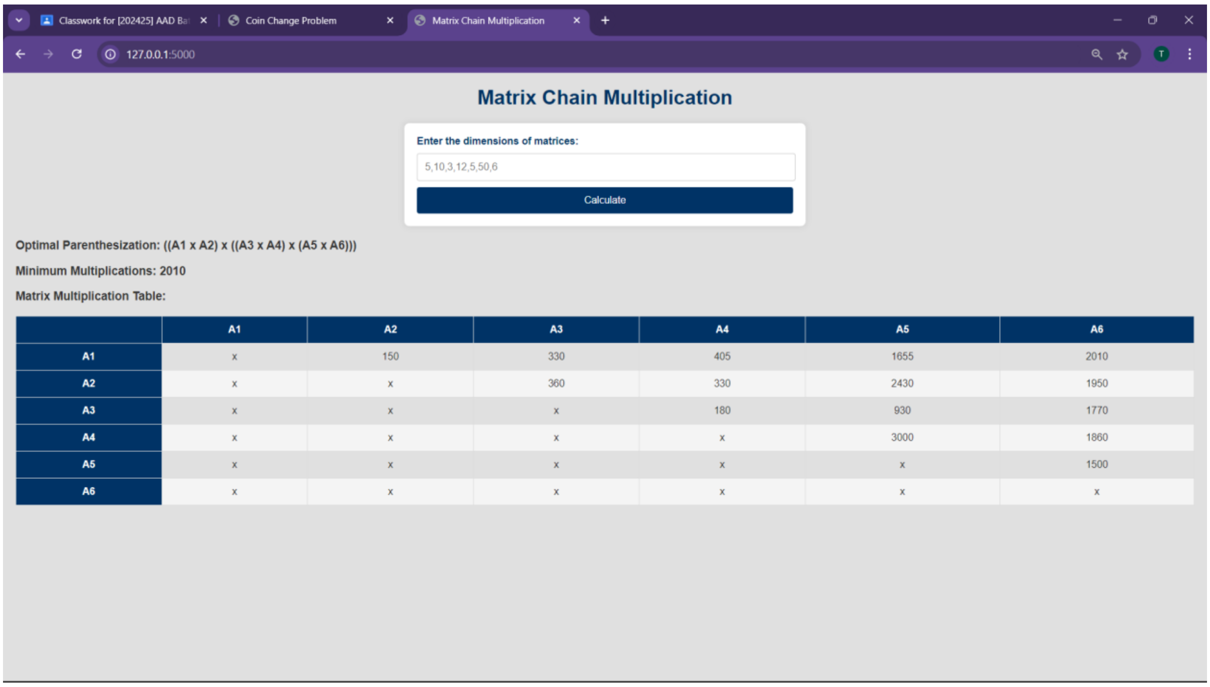
```
from flask import Flask, render_template, request


app = Flask(__name__)


def matrix_chain_order(p):
    n = len(p) - 1
    m = [[0] * n for _ in range(n)]
    s = [[0] * n for _ in range(n)]

    for l in range(2, n + 1):
        for i in range(n - l + 1):
            j = i + l - 1
            m[i][j] = float('inf')

            for k in range(i, j):

                q = m[i][k] + m[k + 1][j] + p[i] * p[k + 1] * p[j + 1]

                if q < m[i][j]:

                    m[i][j] = q

                    s[i][j] = k


    optimal_parens = construct_optimal_solution(s, 0, n - 1)

    return m, optimal_parens


def construct_optimal_solution(s, i, j):

    if i == j:

        return f"A{i+1}"

    else:

        return f"({construct_optimal_solution(s, i, s[i][j])} x {construct_optimal_solution(s, s[i][j] + 1, j)})"


@app.route('/', methods=['GET', 'POST'])

def index():

    result = None


    if request.method == 'POST':

        dimensions = request.form['dimensions']

        p = list(map(int, dimensions.split(',')))

        m, optimal_parens = matrix_chain_order(p)

        min_multiplications = m[0][-1]


        result = {

            'm': m,

            'optimal_parens': optimal_parens,
            'min_multiplications': min_multiplications,
            'dimensions': p
        }
    return render_template('index.html', result=result)


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 Chain Multiplication</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
            background-color: #e0e0e0;
            color: #333;
        }
        h1 {
            color: #003366;
            text-align: center;
        }
        form {
            max-width: 600px;
            margin: 0 auto 20px auto;
            padding: 20px;
            background-color: #ffffff;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }
        label {
            display: block;
            margin-bottom: 10px;
            font-size: 16px;
            font-weight: bold;
            color: #003366;
        }
        input[type="text"] {
            width: calc(100% - 22px);
            padding: 12px;
            border: 1px solid #ccc;
            border-radius: 4px;
            font-size: 16px;
        }
        button {
            padding: 12px 24px;
            border: none;
            border-radius: 4px;
            background-color: #003366;
            color: #ffffff;
            font-size: 16px;
            cursor: pointer;
            margin-top: 10px;
```

```
        display: block;
        width: 100%;
    }
    button:hover {
        background-color: #002244;
    }
    table {
        width: 100%;
        border-collapse: collapse;
        margin: 20px 0;
    }
    th, td {
        padding: 12px;
        border: 1px solid #ddd;
        text-align: center;
        font-size: 16px;
    }
    th {
        background-color: #003366;
        color: #ffffff;
    }
    tr:nth-child(even) {
        background-color: #f4f4f4;
    }
    footer {
        margin-top: 40px;
        font-size: 14px;
        color: #666;
        text-align: center;
        }
    </style>
</head>
<body>
    <h1>Matrix Chain Multiplication</h1>
    <form method="POST">
        <label for="dimensions">Enter the dimensions of matrices:</label>
        <input type="text" name="dimensions" placeholder="5,10,3,12,5,50,6" required>
        <button type="submit">Calculate</button>
    </form>

    {% if result %}
        <h3>Optimal Parenthesization: {{ result.optimal_parens }}</h3>
        <h3>Minimum Multiplications: {{ result.min_multiplications }}</h3>
        <h3>Matrix Multiplication Table:</h3>
        <table>
            <thead>
                <tr>
                    <th></th>
                    {% for j in range(result.dimensions|length - 1) %}
                        <th>A{{ j + 1 }}</th>
                    {% endfor %}
                </tr>
            </thead>
            <tbody>
                {% for i in range(result.dimensions|length - 1) %}
                    <tr>
                        <th>A{{ i + 1 }}</th>
                        {% for j in range(result.dimensions|length - 1) %}
```

```
                            <td>{{ result.m[i][j] if i < j else 'x' }}</td>

                        {% endfor %}

                    </tr>

                {% endfor %}

            </tbody>

        </table>

    {% endif %}


</body>

</html>
```



Matrix Chain Multiplication

Enter the dimensions of matrices:

5,10,3,12,5,50,6

Optimal Parenthesization: ((A1 x A2) x ((A3 x A4) x (A5 x A6)))

Minimum Multiplications: 2010

Matrix Multiplication Table:

	A1	A2	A3	A4	A5	A6
A1	x	150	330	405	1655	2010
A2	x	x	360	330	2430	1950
A3	x	x	x	180	930	1770
A4	x	x	x	x	3000	1860
A5	x	x	x	x	x	1500
A6	x	x	x	x	x	x