# "Experiment 2.1"

Student Name: **SUMIT KUMAR** UID: **20BCS8226**

Branch: **CSE** Section/Group: **808-A**

Semester: **5** Date of Submission: **28-10-22**

Subject Name: **Design and Analysis of Algorithms Lab** Subject Code: **20CSP-312**

## 1. Aim/Overview of the practical:

Code and analyze to find an optimal solution to matrix chain multiplication using dynamic programming.

## 2. Task to be done/ Which logistics used:

Write a program to find the optimal solution of the Matrix Chain Multiplication.

## 3. Requirements (For programming-based labs):

- Laptop or PC.
- Operation system (Mac, Windows, Linux, or any)
- Vs-Code with MinGw or any C++ Compiler

## 4. Algorithm/Flowchart (For programming-based labs)

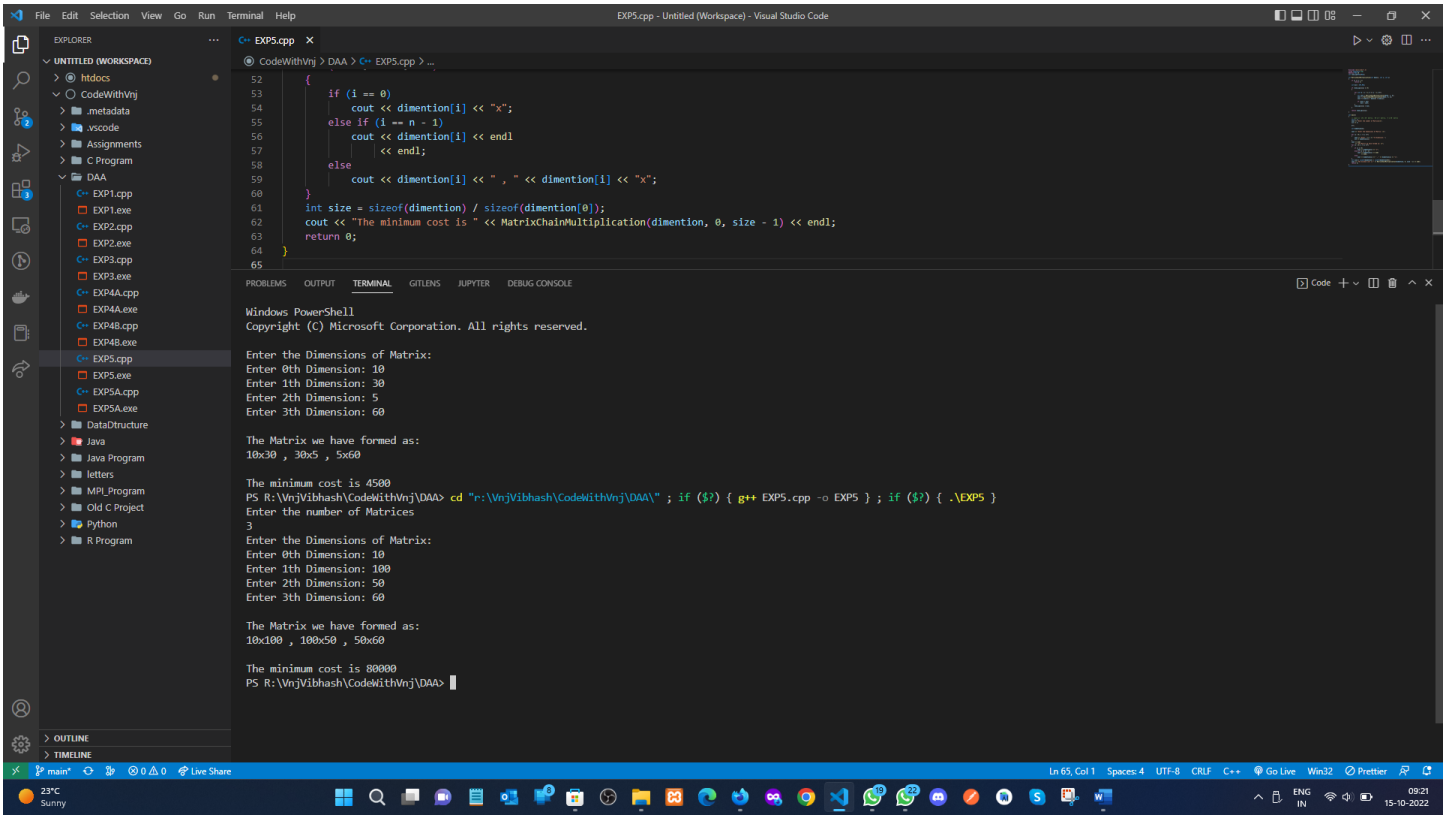
1. First, it will divide the matrix sequence into two subsequences.
2. You will find the minimum cost of multiplying out each subsequence.
3. You will add these costs together and in the price of multiplying the two result matrices.
4. These procedures will be repeated for every possible matrix split and calculate the minimum.

## 5. Steps for experiment/practical/Code:

```
#include <bits/stdc++.h>
using namespace std;
#define MAX 10
int look_up[MAX][MAX];

int MatrixChainMultiplication(int dims[], int i, int j)
{
    if (j <= i + 1)
        return 0;

    int min = INT_MAX;

    if (look_up[i][j] == 0) {

        for (int k = i + 1; k <= j - 1; k++){
            int cost = MatrixChainMultiplication(dims, i, k);
            cost += MatrixChainMultiplication(dims, k, j);
            cost += dims[i] * dims[k] * dims[j];

            if (cost < min)
                min = cost;
        }
        look_up[i][j] = min;
    }

    return look_up[i][j];
}

int main() {
    // input is `10 × 30` matrix, `30 × 5` matrix, `5 × 60` matrix
    int n, i;
    cout << "Enter the number of Matrices\n";
    cin >> n;

    n++;

    int dimention[n];

    cout << "Enter the Dimensions of Matrix: \n";

    for (i = 0; i < n; i++)
    {
```

```
        cout << "Enter " << i << "th Dimension: ";
        cin >> dimention[i];
    }
    cout << endl << "The Matrix we have formed as: \n";
    for (i = 0; i < n; i++)
    {
        if(i==0)
            cout << dimention[i] << "x";
        else if(i==n-1)
            cout << dimention[i] << endl << endl;
        else
            cout << dimention[i] << " , " << dimention[i] << "x";
    }
    int size = sizeof(dimention) / sizeof(dimention[0]);
    cout << "The minimum cost is " << MatrixChainMultiplication(dimention, 0, size - 1) <<
endl;
    return 0;
}
```

## 6. Output:




## Learning outcomes (What I have learnt):

1. How to solve the Matrix Chain Multiplication problem using dynamic programming.
2. How to use the Array elements as a Matrix rows and columns.

## Evaluation Grid (To be created per the faculty's SOP and Assessment guidelines):

| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
|---|---|---|---|
| 1. | Worksheet completion including writing learning objectives/Outcomes. (To be submitted at the end of the day). | | |
| 2. | Post-Lab Quiz Result. | | |
| 3. | Student Engagement in Simulation/Demonstration/Performance and Controls/Pre-Lab Questions. | | |
| | Signature of Faculty (with Date): | Total Marks Obtained: | |