**CHANDIGARH UNIVERSITY**

**UNIVERSITY INSTITUTE OF ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



|  |  |
| --- | --- |
| **Submitted By:** Sahil Kaundal  **Submitted To:** Neha Dutta | |
| **Subject Name** | Design and Analysis of Algorithm Lab |
| **Subject Code** | 20CSP-312 |
| **Branch** | Computer Science Engineering |
| **Semester** | 5th |

**Experiment 5**

**Student Name:** Sahil Kaundal **UID:** 21BCS8197

**Branch:** BE CSE (Lateral Entry) **Section/Group:** 616/A

**Semester:** 5th **Date of Performance:** 14/10/2022

**Subject Name:** DAA Lab **Subject Code:** 21-CSP-312

# Aim/Overview of the practical:

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

# Task to be done/ Which logistics used:

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

# Requirements:

Laptop or PC.

Operation system (Mac, Windows, Linux, or any)

Vs-Code with MinGw or any C++ Compiler

1. **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.

# 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;

}

# Result/Output/Writing Summary:

# 

**Learning outcomes (What I have learnt):**

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

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |