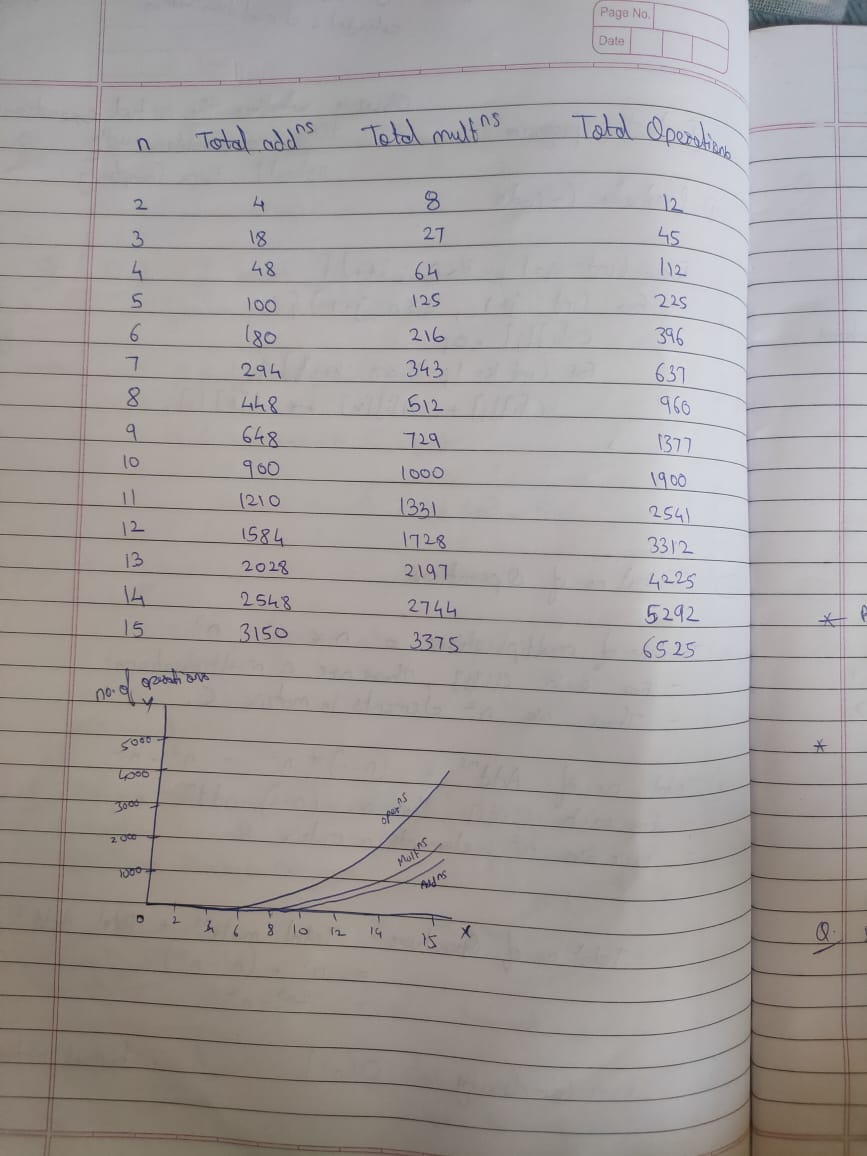
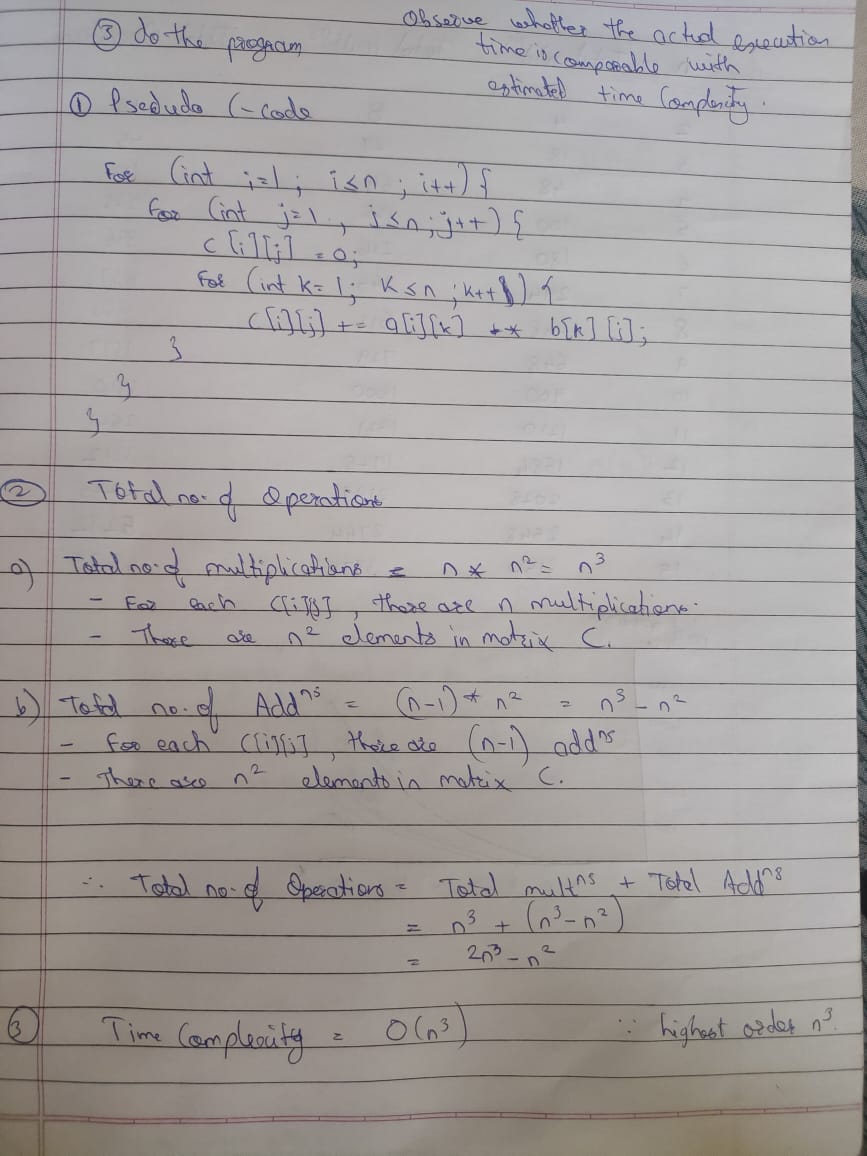
**Experiment No : 1**

**DAAOA LAB**

**Name :** Harshwardhan Patil **ROLL :** 50

**Class :** IT B **Batch :** 1

Problem Statement:   
Time complexity analysis of Square matrix multiplication   
Write pseudo  
Time complexity analysis and estimation graph  
implementation  
Excel plot of actual time taken



|  |  |  |  |
| --- | --- | --- | --- |
| **N (input size)** | **Total\_Additions** | **Total\_Multiplications** | **Total\_Operations** |
| 2 | 4 | 8 | 12 |
| 3 | 18 | 27 | 45 |
| 4 | 48 | 64 | 112 |
| 5 | 100 | 125 | 225 |
| 6 | 180 | 216 | 396 |
| 7 | 294 | 343 | 637 |
| 8 | 448 | 512 | 960 |
| 9 | 648 | 729 | 1377 |
| 10 | 900 | 1000 | 1900 |
| 11 | 1210 | 1331 | 2541 |
| 12 | 1584 | 1728 | 3312 |
| 13 | 2028 | 2197 | 4225 |
| 14 | 2548 | 2744 | 5292 |
| 15 | 3150 | 3375 | 6525 |

1. **Implementation –**

import java.util.Arrays;

public class Main {

    public static void main(String[] args) {

        int n = 3;

        int[][] A = generateMatrix(n);

        int[][] B = generateMatrix(n);

        System.out.println("Matrix A:");

        printMatrix(A);

        System.out.println("Matrix B:");

        printMatrix(B);

        long startTime = System.nanoTime();

        System.out.println("Start Time (ns): " + startTime);

        int[][] C = matrixMultiply(A, B, n);

        long endTime = System.nanoTime();

        System.out.println("End Time (ns): " + endTime);

        System.out.println("\nMatrix C (Result):");

        printMatrix(C);

        double elapsedTime = (endTime - startTime) / 1e6;

        System.out.println("Time taken: " + elapsedTime + " ms");

    }

    public static int[][] generateMatrix(int n) {

        int[][] matrix = new int[n][n];

        for (int i = 0; i < n; i++) {

            for (int j = 0; j < n; j++) {

                matrix[i][j] = (int) (Math.random() \* 10);

            }

        }

        return matrix;

    }

    public static int[][] matrixMultiply(int[][] A, int[][] B, int n) {

        int[][] C = new int[n][n];

        for (int i = 0; i < n; i++) {

            for (int j = 0; j < n; j++) {

                for (int k = 0; k < n; k++) {

                    C[i][j] += A[i][k] \* B[k][j];

                }

            }

        }

        return C;

    }

    public static void printMatrix(int[][] matrix) {

        for (int[] row : matrix) {

            System.out.println(Arrays.toString(row));

        }

        System.out.println();

    }

}

**Output –**

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**A screenshot of a computer

Description automatically generated**

1. **Excel plot of actual time taken –**

|  |  |
| --- | --- |
| n | Time taken (in ms) |
| 3 | 31.08 |
| 4 | 55.95 |
| 5 | 24.55 |
| 6 | 35.55 |