Walchand College Of Engineering

***Department of Computer Science***

**Batch: T3**

**Assignment No. 4**

**Divide and conquer strategy (Part 2)**

**Name: Piyush Dawkhare**

**PRN: 23510056**

**Problem Statement 1**:

Implement the naive method to multiply two n × n matrices and justify why the time complexity is O(n³).

# Algorithms Used

* Naive Triple-Loop Matrix Multiplication

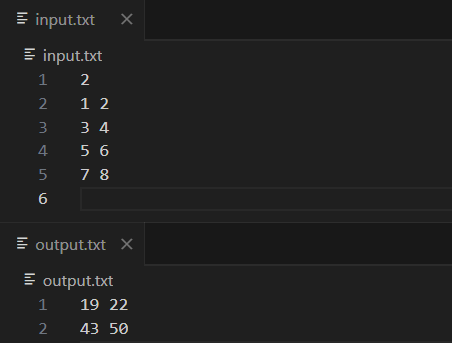
# Related Theory

* Requires n³ multiplications/additions.
* Time Complexity: O(n³)
* Space Complexity**: O(n²)**

**Algorithm / Procedure**

1. Initialize result matrix C with zeros.
2. For each row i in A and column j in B:
   * Multiply row elements of A with column elements of B.
   * Sum them into C[i][j].
3. Print final matrix C.

# Output

****

**Problem Statement 2**:

Implement Strassen’s Matrix Multiplication for 3×3 matrices and analyze the time complexity.

# Algorithms Used

* Strassen’s Divide and Conquer Matrix Multiplication

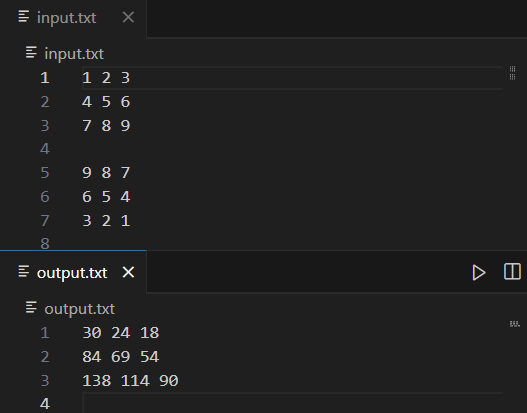
# Related Theory

* Strassen reduces the 8 multiplications of 2×2 block matrices to 7 multiplications + additions/subtractions.
* Recursively applied, this reduces complexity from:
  + Naive: O(n³)
  + Strassen: O(n^log₂7) ≈ O(n^2.81)
* This gives a faster method for large matrices.

# Algorithm / Procedure

* Pad 3×3 matrices to 4×4.
* Split each into four submatrices (A11, A12, A21, A22).
* Compute 7 products (M1 … M7).
* Combine to get result submatrices (C11, C12, C21, C22).
* Output only 3×3 part.

**Output**

****