170) Given two 2×2 Matrices A and B

A=(1 7 B=(1 3

3 5) 7 5)

Use Strassen's matrix multiplication algorithm to compute the product matrix C such that C=A×B.

Test Cases:

Consider the following matrices for testing your implementation:

Test Case 1:

A=(1 7 B=(6 8
3 5), 4 2)

Expected Output:
C=(18 14
62 66)

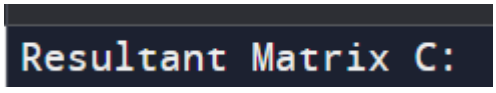
AIM: TO write a python program for Given two 2×2 Matrices A and B

PROGRAM :

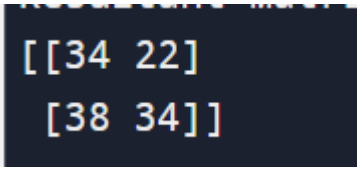
```
import numpy as np


def strassen_multiply(A, B):
    # Base case: when the matrices are 1x1
    if len(A) == 1:
        return A * B

    a11, a12, a21, a22 = A[0, 0], A[0, 1], A[1, 0], A[1, 1]
    b11, b12, b21, b22 = B[0, 0], B[0, 1], B[1, 0], B[1, 1]

    p1 = (a11 + a22) * (b11 + b22)
    p2 = (a21 + a22) * b11
```

```
    p3 = a11 * (b12 - b22)

    p4 = a22 * (b21 - b11)

    p5 = (a11 + a12) * b22

    p6 = (a21 - a11) * (b11 + b12)

    p7 = (a12 - a22) * (b21 + b22)


    # Combine the 7 products to get the final 2x2 result matrix

    c11 = p1 + p4 - p5 + p7

    c12 = p3 + p5

    c21 = p2 + p4

    c22 = p1 + p3 - p2 + p6


    # Construct the resulting matrix

    C = np.array([[c11, c12], [c21, c22]])


    return C


A = np.array([[1, 7], [3, 5]])

B = np.array([[6, 8], [4, 2]])


C = strassen_multiply(A, B)

print("Resultant Matrix C:")

print(C)
```

INPUT:



OUTPUT:



TIME COMPLEXITY: O(n^3)