3.15 STRASSEN'S MATRIX MULTIPLICATION FOR 2×2 MATRICES

Question:

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

AIM

To compute the product of two 2×2 matrices using Strassen's algorithm, which reduces the number of multiplications for improved efficiency

ALGORITHM

1. Compute seven intermediate products (P1 to P7):
    - P1 = a × (f − h)
    - P2 = (a + b) × h
    - P3 = (c + d) × e
    - P4 = d × (g − e)
    - P5 = (a + d) × (e + h)
    - P6 = (b − d) × (g + h)
    - P7 = (a − c) × (e + f)
2. Use these products to calculate the four elements of the result matrix C:
    - Top-left element (C0][0]) = P5 + P4 − P2 + P6
    - Top-right element (C0][1]) = P1 + P2
    - Bottom-left element (C1][0]) = P3 + P4
    - Bottom-right element (C1][1]) = P1 + P5 − P3 − P7
3. Assemble the result matrix C using these four computed values.

PROGRAM

```
def strassen_2x2(A, B):
    a, b, c, d = A[0][0], A[0][1], A[1][0], A[1][1]
    e, f, g, h = B[0][0], B[0][1], B[1][0], B[1][1]

    p1 = a * (f - h)
    p2 = (a + b) * h
    p3 = (c + d) * e
    p4 = d * (g - e)
    p5 = (a + d) * (e + h)
    p6 = (b - d) * (g + h)
    p7 = (a - c) * (e + f)

    return [
        [p5 + p4 - p2 + p6, p1 + p2],
        [p3 + p4, p1 + p5 - p3 - p7]
    ]

def run_strassen():
    A = [[int(x) for x in input("Enter row 1 of A: ").split()],
         [int(x) for x in input("Enter row 2 of A: ").split()]]
    B = [[int(x) for x in input("Enter row 1 of B: ").split()],
         [int(x) for x in input("Enter row 2 of B: ").split()]]
    C = strassen_2x2(A, B)
    print("Product matrix:")
    for row in C:
        print(*row)

run_strassen()
```

Input:

A= 1 5(row 1), 6 9(row 2)

B = 11 3(row 1), 8 2(row 2)

Output:

```
Enter row 1 of A: 1 5
Enter row 2 of A: 6 9
Enter row 1 of B: 11 3
Enter row 2 of B: 8 2
Product matrix:
51 13
138 36
>>>
```

RESULT:

Thus the program is successfully executed and the output is verified.

PERFORMANCE ANALYSIS:

- Time Complexity: $O(n^3)$
- Space Complexity: $O(n^2)$