DAA EXP3

Name: Rahul Chalwadi
CSE-DS(D1 Batch)
UID: 2021700012

AIM: EXPERIMENT BASED ON DIVIDE AND CONQUER APPROACH TO IMPLEMENT STRASSEN'S MATRIX MULTIPLICATION.

Theory:

Strassen's Matrix Multiplication Algorithm

In this context, using Strassen's Matrix multiplication algorithm, the time consumption can be improved a little bit.

Strassen's Matrix multiplication can be performed only on **square matrices** where **n** is a **power of 2**. Order of both of the matrices are **n × n**.

Divide **X**, **Y** and **Z** into four (n/2)×(n/2) matrices as represented below −

$$Z = \begin{bmatrix} I & J \\ K & L \end{bmatrix} \qquad X = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \text{ and } \qquad Y = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

Using Strassen's Algorithm compute the following −

$$M_1 := (A + C) \times (E + F)$$

$$M_2 := (B + D) \times (G + H)$$

$$M_2 := (B + D) \times (G + H)$$

$$M_3 := (A - D) \times (E + H)$$

$$M_4 := A \times (F - H)$$

$$M_5 := (C + D) \times (E)$$

$$M_6 := (A + B) \times (H)$$

$$M_7 := D \times (G - E)$$

Then,

$$I := M_2 + M_3 - M_6 - M_7$$

$$J := M_4 + M_6$$

$$K := M_5 + M_7$$

$$L := M_1 - M_3 - M_4 - M_5$$

Analysis

$$T(n) = \begin{cases} c & if\, n = 1 \\ 7\, x\, T(\frac{n}{2}) + d\, x\, n^2 & otherwise \end{cases}$$ where *c* and *d* are constants

Using this recurrence relation, we get $T(n) = O(n^{log7})$

Hence, the complexity of Strassen's matrix multiplication algorithm is $O(n^{log7})$.

CODE:

```
#include<stdio.h>
int main(){
  int a[2][2], b[2][2], c[2][2], i, j;
  int m1, m2, m3, m4 , m5, m6, m7;

  printf("Enter the 4 elements of first matrix: ");
  for(i = 0;i < 2; i++)
      for(j = 0;j < 2; j++)
           scanf("%d", &a[i][j]);

  printf("Enter the 4 elements of second matrix: ");
  for(i = 0; i < 2; i++)
      for(j = 0;j < 2; j++)
           scanf("%d", &b[i][j]);

  printf("\nThe first matrix is\n");
  for(i = 0; i < 2; i++){
      printf("\n");
      for(j = 0; j < 2; j++)
           printf("%d\t", a[i][j]);
  }

  printf("\nThe second matrix is\n");
  for(i = 0;i < 2; i++){
      printf("\n");
      for(j = 0;j < 2; j++)
           printf("%d\t", b[i][j]);
  }

  m1= (a[0][0] + a[1][1]) * (b[0][0] + b[1][1]);
  m2= (a[1][0] + a[1][1]) * b[0][0];
  m3= a[0][0] * (b[0][1] - b[1][1]);
```

```
 m4= a[1][1] * (b[1][0] - b[0][0]);
 m5= (a[0][0] + a[0][1]) * b[1][1];
 m6= (a[1][0] - a[0][0]) * (b[0][0]+b[0][1]);
 m7= (a[0][1] - a[1][1]) * (b[1][0]+b[1][1]);

 c[0][0] = m1 + m4- m5 + m7;
 c[0][1] = m3 + m5;
 c[1][0] = m2 + m4;
 c[1][1] = m1 - m2 + m3 + m6;

  printf("\n Strassen's algorithm used for multiplication\n");
  for(i = 0; i < 2 ; i++){
    printf("\n");
    for(j = 0;j < 2; j++)
        printf("%d\t", c[i][j]);
  }

  return 0;
}
```

OUTPUT:

```
Enter the 4 elements of first matrix: 1 2 3 4
Enter the 4 elements of second matrix: 5 6 7 8
The first matrix is

1   2
3   4
The second matrix is

5   6
7   8
 Strassen's algorithm used for multiplication

19  22
43  50
```

Conclusion: In this experiment, I understood about the strassen algorithm and implemented it in C..