Experiment No.	**3**
Name	**Varada Khadake**
UID No.	**2021300059**
Class & Division	**COMPS A BATCH D**

Aim:To implement strassens multiplication.

Observation/Theory:

Strassen's Matrix Multiplication Algorithm

In this context, using Strassen's Matrix multiplication algorithm, the time consumption car 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 −

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

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

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

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

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

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

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

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 $O(n^{log7})$.

Algorithm:

```
begin
          If n = threshold then compute
                    C = a * b is a conventional matrix.
          Else
                    Partition a into four sub matrices  a11, a12, a21, a22.
                    Partition b into four sub matrices b11, b12, b21, b22.
                    Strassen ( n/2, a11 + a22, b11 + b22, d1)
                    Strassen ( n/2, a21 + a22, b11, d2)
                    Strassen ( n/2, a11, b12 − b22, d3)
                    Strassen ( n/2, a22, b21 − b11, d4)
                    Strassen ( n/2, a11 + a12, b22, d5)
                    Strassen (n/2, a21 − a11, b11 + b22, d6)
                    Strassen (n/2, a12 − a22, b21 + b22, d7)

                    C = d1+d4-d5+d7     d3+d5
                    d2+d4           d1+d3-d2-d6

          end if

          return (C)
end.
```

Code:

```c
#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("\nThe first matrix is\n");
    for (i = 0; i < 2; i++)
    {
        printf("\n");
        for (j = 0; j < 2; j++)
        {
            a[i][j] = (rand() % 100) + 1;
            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++)
        {
            b[i][j] = (rand() % 100) + 1;
            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("\nAfter multiplication using Strassen's algorithm \n");
    for (i = 0; i < 2; i++)
    {
        printf("\n");
        for (j = 0; j < 2; j++)
            printf("%d\t", c[i][j]);
    }

    return 0;
}
```

Output:

```
PS C:\Users\varad\OneDrive\Documents\sem 4\DAA\DAA lab code> cd "c:\Users\varad\OneDrive\Documents\sem 4\DAA\DAA l
ab code\" ; if ($?) { gcc strassen.c -o strassen } ; if ($?) { .\strassen }
strassen.c: In function 'main':
strassen.c:29:24: warning: implicit declaration of function 'rand' [-Wimplicit-function-declaration]
             a[i][j] = (rand() % 100) + 1;
                        ^~~~

The first matrix is

42      68
35      1
The second matrix is

70      25
79      59
After multiplication using Strassen's algorithm

8312    5062
2529    934
PS C:\Users\varad\OneDrive\Documents\sem 4\DAA\DAA lab code> 
```

Conclusion: I implemented stassen’s multiplication method and understood time complexity it takes. Also, I took random numbers as matrix and performed multiplication.