Name	MAAZ SHAIKH
UID no.	2021700059
Experiment No.	03

AIM:	**Divide and Conquer- Strassen's Matrix Multiplication.**
ALGORITHM/ THEORY:	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 − $Z = \begin{bmatrix} I & J \\ K & L \end{bmatrix}$ $X = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ and $Y = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$

	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 $O(n^{log7})$.

PROGRAM:	(code below)

```
#include<stdio.h>
int main(){

  int i,j;
  int p,q,r,s,t,u,v;
  int a[2][2],b[2][2],c[2][2];

  printf("\n\t\t*******Strassins Matrix Multiplication
Program*******\t\t\n");

  printf("\nPlease 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("Please 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]);
  }
  //Following are the strassins formulas
  p= (a[0][0] + a[1][1])*(b[0][0]+b[1][1]);
  q= (a[1][0]+a[1][1])*b[0][0];
  r= a[0][0]*(b[0][1]-b[1][1]);
  s= a[1][1]*(b[1][0]-b[0][0]);
  t= (a[0][0]+a[0][1])*b[1][1];
  u= (a[1][0]-a[0][0])*(b[0][0]+b[0][1]);
  v= (a[0][1]-a[1][1])*(b[1][0]+b[1][1]);
```

	`` c[0][0]=p+s-t+v; c[0][1]=r+t; c[1][0]=q+s; c[1][1]=p-q+r+s; printf("\nMatrix multiplication using Strassins Algorithm: \n"); for(i=0;i<2;i++){ printf("\n"); for(j=0;j<2;j++) printf("%d\t",c[i][j]); } return 0; } ```

RESULT:

```
                *******Strassins Matrix Multiplication Program*******

Please enter the 4 elements of first matrix: 4 5 3 1
Please Enter the 4 elements of second matrix: 7 8 9 1

The first matrix is

4       5
3       1
The second matrix is

7       8
9       1
Matrix multiplication using Strassins Algorithm:

73      37
30      42
PS C:\Users\maazs\OneDrive\Desktop\Studies\DAA\DAA Coding>
```

CONCLUSION:	I understood the Strassen’s matrix multiplication algorithm and also understood how its time complexity is less than normal matrix multiplication method.