

NAME:	Harshal Chawan
UID:	2021300019
SUBJECT	Design and Analysis of Algorithm
EXPERIMENT NO :	03
DATE OF PERFORMANCE	27/02/2023
DATE OF SUBMISSION	6/03/2023
AIM:	To multiply two matrices using strassen's matrix multiplication.
PROBLEM STATEMENT 1:	Strassen's matrix multiplication on a generalized 2x2 matrix.
ALGORITHM and THEORY:	<p>Strassen in 1969 which gives an overview that how we can find the multiplication of two 2*2 dimension matrix by the brute-force algorithm. But by using divide and conquer technique the overall complexity for multiplication two matrices is reduced. This happens by decreasing the total number of multiplication performed at the expenses of a slight increase in the number of addition.</p> <p>For multiplying the two 2*2 dimension matrices Strassen's used some formulas in which there are seven multiplication and eighteen addition, subtraction, and in brute force algorithm, there is eight multiplication and four addition. The utility of Strassen's formula is shown by its asymptotic superiority when order n of matrix reaches infinity. Let us consider two matrices A and B, n*n dimension, where n is a power of two. It can be observed that we can contain four n/2*n/2 submatrices from A, B and their product C. C is the resultant matrix of A and B.</p> <p>Procedure of Strassen matrix multiplication</p> <p>There are some procedures:</p> <ol style="list-style-type: none"> 1. Divide a matrix of order of 2*2 recursively till we get the matrix of

2*2.

2. Use the previous set of formulas to carry out 2*2 matrix multiplication.
3. In this eight multiplication and four additions, subtraction are performed.
4. Combine the result of two matrixes to find the final product or final matrix.

Formulas for Stassen's matrix multiplication

In **Strassen's matrix multiplication** there are seven multiplication and four addition, subtraction in total.

1. $D1 = (a11 + a22) (b11 + b22)$
2. $D2 = (a21 + a22) .b11$
3. $D3 = (b12 - b22) .a11$
4. $D4 = (b21 - b11) .a22$
5. $D5 = (a11 + a12) .b22$
6. $D6 = (a21 - a11) . (b11 + b12)$
7. $D7 = (a12 - a22) . (b21 + b22)$

$$C11 = d1 + d4 - d5 + d7$$

$$C12 = d3 + d5$$

$$C21 = d2 + d4$$

$$C22 = d1 + d3 - d2 - d6$$

PROGRAM:

```
#include<stdio.h>
#include<time.h>
void main()
{
    int a[2][2],b[2][2],c[2][2],i,j;
    int p[7];
    int s[10];
    clock_t start,end;
    printf("Enter the elements of 1st matrix:");
    for(i=0;i<2;i++)
    {
        for(j=0;j<2;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    printf("Enter the elements of 2nd matrix:");
```

```
for(i=0;i<2;i++)
{
    for(j=0;j<2;j++)
    {
        scanf("%d",&b[i][j]);
    }
}
start=clock();
s[0]=b[0][1]-b[1][1];
s[1]=a[0][0]+a[0][1];
s[2]=a[1][0]+a[1][1];
s[3]=b[1][0]-b[0][0];
s[4]=a[0][0]+a[1][1];
s[5]=b[0][0]+b[1][1];
s[6]=a[0][1]-a[1][1];
s[7]=b[1][0]+b[1][1];
s[8]=a[0][0]-a[1][0];
s[9]=b[0][0]+b[0][1];

p[0]=s[0]*a[0][0];
p[1]=s[1]*b[1][1];
p[2]=s[2]*b[0][0];
p[3]=s[3]*a[1][1];
p[4]=s[4]*s[5];
p[5]=s[6]*s[7];
p[6]=s[8]*s[9];

c[0][0]=p[4]+p[3]-p[1]+p[5];
c[0][1]=p[0]+p[1];
c[1][0]=p[2]+p[3];
c[1][1]=p[4]+p[0]-p[2]-p[6];

for(i=0;i<10;i++)
{
```

```

        printf("\nS%d=%d ",i+1,s[i]);
    }
    printf("\n");
    for(j=0;j<7;j++)
    {
        printf("\np%d=%d ",j+1,p[j]);
    }
    printf("\n\n");
    printf("MATRIX A:-\n");
    for(i=0;i<2;i++)
    {
        printf("\n");
        for(j=0;j<2;j++)
        {
            printf("%d\t",a[i][j]);
        }
    }
    printf("\n");
    printf("MATRIX B:-\n");
    for(i=0;i<2;i++)
    {
        printf("\n");
        for(j=0;j<2;j++)
        {
            printf("%d\t",b[i][j]);
        }
    }
    printf("\n");
    printf("MATRIX C:-\n\n");
    printf("%d\t%d\n%d\t%d\n",c[0][0],c[0][1],c[1][0],c[1][1]);
    end=clock();
    printf("The time taken by the program: ");
    printf("%lf",(double)(end-start)/CLOCKS_PER_SEC);
}

```

OUTPUT:

```
students@students-HP-280-G3-MT:~$ ./a.out
Enter the elements of 1st matrix:1 3 7 5
Enter the elements of 2nd matrix:6 8 4 2

S1=6
S2=4
S3=12
S4=-2
S5=6
S6=8
S7=-2
S8=6
S9=-6
S10=14

p1=6
p2=8
p3=72
p4=-10
p5=48
p6=-12
p7=-84

MATRIX A:-

1      3
7      5
MATRIX B:-

6      8
4      2
MATRIX C:-

18     14
62     66
The time taken by the program: 0.000209students@students-HP-280-G3-MT:~$
```

CONCLUSION:

By performing above experiment I have understood how the time complexity of strassen's matrix multiplication is better than that of normal $m \times n$ matrix multiplication.