Sardar Patel Institute of Technology

SEM IV: DESIGN AND ANALYSIS OF ALGORITHMS.

Name	SMRUTI SONEKAR
UID no.	2021700064
BRANCH:	SE CSE (DATA SCIENCE)
Experiment No.	3

TOPIC:	STRASSEN'S MATRIX MULTIPLICATION.
THEORY:	• In normal matrix multiplication of 2X2 we use multiplication function 8 times, so by master theorem, • T(n) = 8T(n/2) + O (n^2), so a= 8,b=2 , n^(logb(a)) = n^(log2(8)) = n^3 , so time complexity of normal matrix multiplication is O(n^3). • In Strassen's method it uses divide and conquer approach. • It is to reduce the number of recursive calls to 7. The sense that this method also divide matrices to sub-matrices of size N/2 x N/2 as shown in the above diagram, but in Strassen's method, the four sub-matrices of result are calculated using following formulae. • T(n) = 7T (n/2) + O(n^2) , so a= 7,b=2 , n^(logb(a)) = n^(log2(7)) = n^2.81 , so time complexity of Strassen's matrix multiplication is O(n^2.81)., it relatively less than normal multiplication.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}_{A} \times \begin{bmatrix} e & f \\ g & h \end{bmatrix}_{B} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}_{C}$$

- A, B and C are square metrices of size N x N
  a, b, c and d are submatrices of A, of size N/2 x N/2
  e, f, g and h are submatrices of B, of size N/2 x N/2

PROGRAM:

```
#include <stdio.h>

int main()
{

    int A[3][3],B[3][3],C[3][3],i,j;
    int p,q,r,s,t,u,v;

    printf("\nEnter numbers for matrix 2x2 A:\n");
    for(i=1;i<=2;i++)
    {
        for(j=1;j<=2;j++)
        {
            scanf("%d",&A[i][j]);
            }
    }

        printf(" \nA matrix is:\n");
    for(i=1;i<=2;i++)
    {
        printf("\n");
        for(j=1;j<=2;j++)
        {
            printf("\t%d \t",A[i][j]);
            }
    }

        printf("\nEnter numbers for matrix 2x2 B:\n");
    for(i=1;i<=2;i++)
    {
        for(j=1;j<=2;j++)
        {
            scanf("%d",&B[i][j]);
            }
```

	(code below)

```
    }

         printf("\n B matrix is:\n");
    for(i=1;i<=2;i++)
    {
        printf("\n");
        for(j=1;j<=2;j++)
        {
            printf("\t%d \t",B[i][j]);
            }
    }

    p= (A[1][1] + A[2][2]) * (B[1][1]+B[2][2]);
    q= (A[2][1] + A[2][2]) *B[1][1];
    r=A[1][1] * (B[1][2]-B[2][2]);
    s= A[2][2] * (B[2][1]-B[1][1]);
    t= (A[1][1] + A[1][2])*B[2][2];
    u= (A[2][1] - A[1][1]) * (B[1][1]+B[1][2]);
    v= (A[1][2] - A[2][2]) * (B[2][1]+B[2][2]);

    C[1][1] = p+s-t+v;
    C[1][2] = r+t;
    C[2][1] = q+s;
    C[2][2] = p+r-q+u;

        printf("\nAnswer for C matrix:\n");
    for(i=1;i<=2;i++)
    {
         printf("\n");
        for(j=1;j<=2;j++)
        {
            printf("\t%d \t",C[i][j]);
            }
    }

    return 0;
}
```

RESULT:

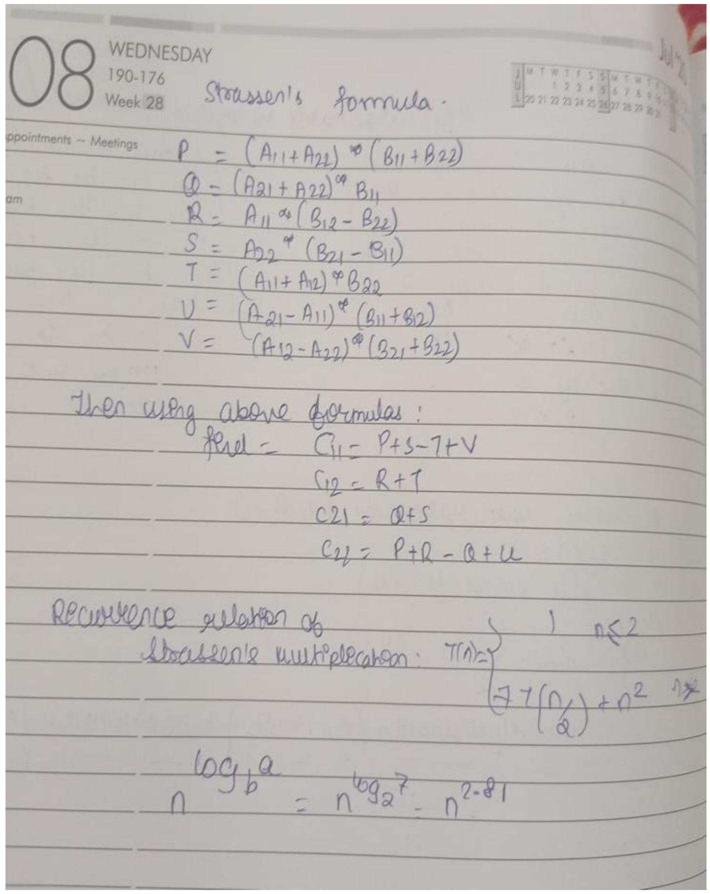
Strassen's Matrix Multiplication :

```
PS D:\c_programming\mudir\daa> gcc matrix.c
PS D:\c_programming\mudir\daa> .\a.exe

Enter numbers for matrix 2x2 A:
1 2
A matrix is:

        1               2
        3               4
Enter numbers for matrix 2x2 B:
5 6
7 8

 B matrix is:

        5               6
        7               8
Answer for C matrix:

        19              22
        43              50
```

```
PS D:\c_programming\mudir\daa> gcc matrix.c
PS D:\c_programming\mudir\daa> .\a.exe

Enter numbers for matrix 2x2 A:
10 20
-2 4

A matrix is:

        10              20
        -2              4
Enter numbers for matrix 2x2 B:
-5 -1
18 3

 B matrix is:

        -5              -1
        18              3
Answer for C matrix:

        310             50
        82              14
```

SOLVED ON PAPER:

Eg 1]



08 WEDNESDAY
190-176
Week 28

Appointments ~ Meetings

Strassen's formula.

$P = (A_{11}+A_{22}) * (B_{11}+B_{22})$

$Q = (A_{21}+A_{22}) * B_{11}$

$R = A_{11} * (B_{12}-B_{22})$

$S = A_{22} * (B_{21}-B_{11})$

$T = (A_{11}+A_{12}) * B_{22}$

$U = (A_{21}-A_{11}) * (B_{11}+B_{12})$

$V = (A_{12}-A_{22}) * (B_{21}+B_{22})$

Then using above formulas:
find = $C_{11} = P+S-T+V$

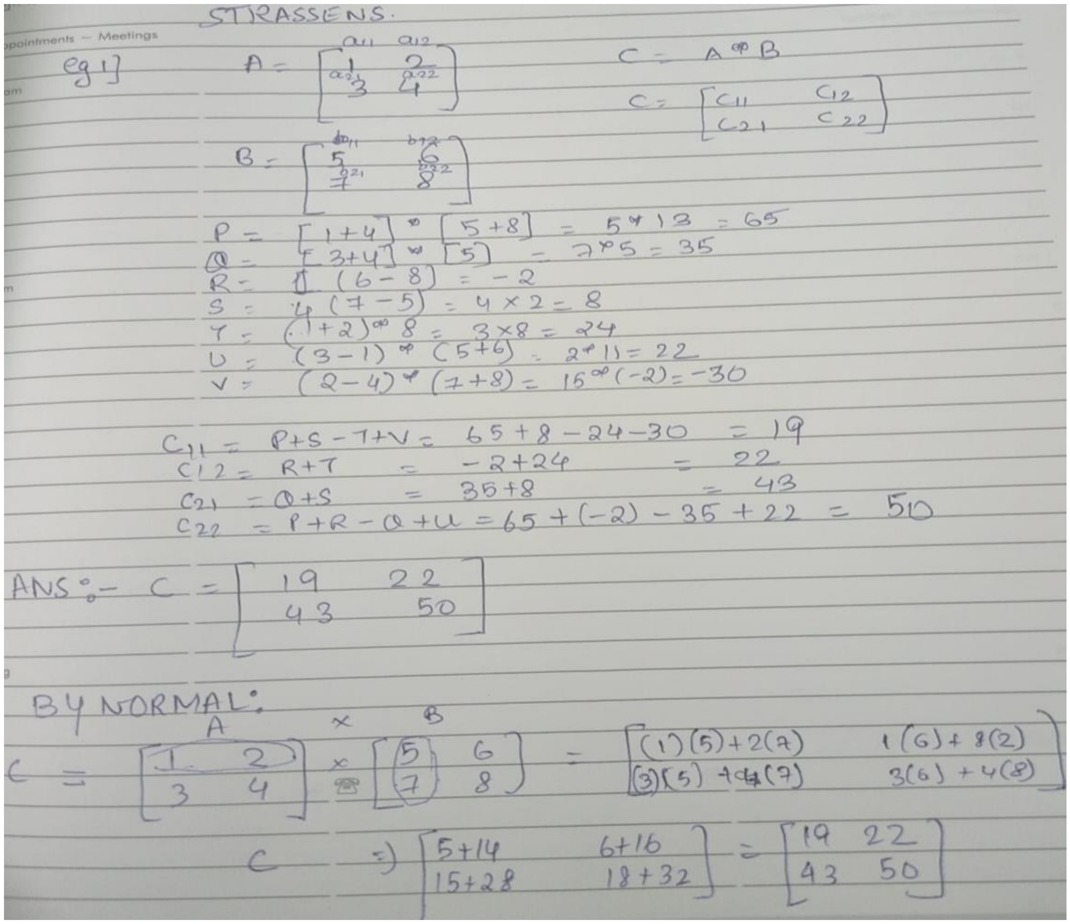
$C_{12} = R+T$

$C_{21} = Q+S$

$C_{22} = P+R-Q+U$

Recurrence relation of Strassen's multiplication: $T(n) = \begin{cases} 1 & n \leq 2 \\ 7T\left(\frac{n}{2}\right)+n^2 & n > 2 \end{cases}$

$n^{\log_b a} = n^{\log_2 7} = n^{2.81}$

<table>
<tr>
<td></td>
<td>


STRASSENS.

eg 1]

$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ $\quad C = A * B \quad C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$

$B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$

$P = [1+4] * [5+8] = 5*13 = 65$

$Q = [3+4] * [5] = 7*5 = 35$

$R = 1\ (6-8) = -2$

$S = 4\ (7-5) = 4 \times 2 = 8$

$T = (1+2) * 8 = 3 \times 8 = 24$

$U = (3-1) * (5+6) = 2*11 = 22$

$V = (2-4) * (7+8) = 15 * (-2) = -30$

$C_{11} = P+S-T+V = 65+8-24-30 = 19$

$C_{12} = R+T = -2+24 = 22$

$C_{21} = Q+S = 35+8 = 43$

$C_{22} = P+R-Q+U = 65+(-2)-35+22 = 50$

ANS :- $C = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$

BY NORMAL:

$C = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} (1)(5)+2(7) & 1(6)+8(2) \\ (3)(5)+4(7) & 3(6)+4(8) \end{bmatrix}$

$C \Rightarrow \begin{bmatrix} 5+14 & 6+16 \\ 15+28 & 18+32 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$

Practically Strassen's method takes a lot of time and it is not applicable for practical uses:

1. **The constants used in Strassen's method are high and for a typical application Naive method works better.**
2. **For Sparse matrices, there are better methods especially designed for them.**
3. **The submatrices in recursion take extra space.**
4. **Because of the limited precision of computer arithmetic on non-integer values, larger errors accumulate in Strassen's algorithm than in Naive Method.**
</td>
</tr>
<tr>
<td>CONCLUSION:</td>
<td>Strassen's method divide and conquer concept and its functionality is applied through program.</td>
</tr>
</table>