G. H. RAISONI COLLEGE OF ENGG., NAGPUR

(An Autonomous Institute under UGC Act 1956)

Department of Computer Science & Engg.

Date: 21/07/2020

Practical Subject: Design and Analysis of Algorithms
Session: 2020-21

Student Details:

Roll Number	49
Name	Shivam Tawari
Semester	3
Section	A
Branch	Artificial Intelligence

Practical Details: Practical Number- 03

Practical Aim	To Implement and analyze time complexity of algorithm using Divide and Conquer Technique.
Theory & Algorithm	***Theory:*** In linear algebra, the Strassen algorithm, named after Volker Strassen, is an algorithm for matrix multiplication. It is faster than the standard matrix multiplication algorithm and is useful in practice for large matrices, but would be slower than the fastest known algorithms for extremely large matrices. Let A, B be two square matrices over a ring R. We want to calculate the matrix product C as $\mathbf{C} = \mathbf{A}\mathbf{B} \qquad \mathbf{A}, \mathbf{B}, \mathbf{C} \in R^{2^n \times 2^n}$ If the matrices A, B are not of type 2n × 2n we fill the missing rows and columns with zeros. We partition A, B and C into equally sized block matrices

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \mathbf{C}_{1,1} & \mathbf{C}_{1,2} \\ \mathbf{C}_{2,1} & \mathbf{C}_{2,2} \end{bmatrix}$$

With

$$\mathbf{A}_{i,j}, \mathbf{B}_{i,j}, \mathbf{C}_{i,j} \in R^{2^{n-1} \times 2^{n-1}}$$

The naive algorithm would be:

$$\begin{aligned} \mathbf{C}_{1,1} &= \mathbf{A}_{1,1}\mathbf{B}_{1,1} + \mathbf{A}_{1,2}\mathbf{B}_{2,1} \\ \mathbf{C}_{1,2} &= \mathbf{A}_{1,1}\mathbf{B}_{1,2} + \mathbf{A}_{1,2}\mathbf{B}_{2,2} \\ \mathbf{C}_{2,1} &= \mathbf{A}_{2,1}\mathbf{B}_{1,1} + \mathbf{A}_{2,2}\mathbf{B}_{2,1} \\ \mathbf{C}_{2,2} &= \mathbf{A}_{2,1}\mathbf{B}_{1,2} + \mathbf{A}_{2,2}\mathbf{B}_{2,2} \end{aligned}$$

With this construction we have not reduced the number of multiplications. We still need 8 multiplications to calculate the Ci,j matrices, the same number of multiplications we need when using standard matrix multiplication.

The Strassen algorithm defines instead new matrices:

$$\begin{aligned} \mathbf{M}_1 &:= (\mathbf{A}_{1,1} + \mathbf{A}_{2,2})(\mathbf{B}_{1,1} + \mathbf{B}_{2,2}) \\ \mathbf{M}_2 &:= (\mathbf{A}_{2,1} + \mathbf{A}_{2,2})\mathbf{B}_{1,1} \\ \mathbf{M}_3 &:= \mathbf{A}_{1,1}(\mathbf{B}_{1,2} - \mathbf{B}_{2,2}) \\ \mathbf{M}_4 &:= \mathbf{A}_{2,2}(\mathbf{B}_{2,1} - \mathbf{B}_{1,1}) \\ \mathbf{M}_5 &:= (\mathbf{A}_{1,1} + \mathbf{A}_{1,2})\mathbf{B}_{2,2} \\ \mathbf{M}_6 &:= (\mathbf{A}_{2,1} - \mathbf{A}_{1,1})(\mathbf{B}_{1,1} + \mathbf{B}_{1,2}) \\ \mathbf{M}_7 &:= (\mathbf{A}_{1,2} - \mathbf{A}_{2,2})(\mathbf{B}_{2,1} + \mathbf{B}_{2,2}) \end{aligned}$$

only using 7 multiplications (one for each Mk) instead of 8. We may now express the Ci,j in terms of Mk:

$$\begin{aligned} \mathbf{C}_{1,1} &= \mathbf{M}_1 + \mathbf{M}_4 - \mathbf{M}_5 + \mathbf{M}_7 \\ \mathbf{C}_{1,2} &= \mathbf{M}_3 + \mathbf{M}_5 \\ \mathbf{C}_{2,1} &= \mathbf{M}_2 + \mathbf{M}_4 \\ \mathbf{C}_{2,2} &= \mathbf{M}_1 - \mathbf{M}_2 + \mathbf{M}_3 + \mathbf{M}_6 \end{aligned}$$

We iterate this division process n times (recursively) until the submatrices degenerate into numbers (elements of the ring R). The resulting product will be padded with zeroes just like A and B, and should be stripped of the corresponding rows and columns.

	Algorithm: **Step 1:** Start **Step 2:** if n is threshold then compute C = a*b else go to Step 3 **Step 3:** Partition ‘A’ into four sub matrices a11, a12, a21, a22 **Step 4:** Partition ‘B’ into four sub matrices b11, b12, b21, b22 **Step 5:** Recursively call Strassen (n/2, a11 + a22, b11 + b22, d1) **Step 6:** Recursively call Strassen (n/2, a21 + a22, b11, d2) **Step 7:** Recursively call Strassen (n/2, a11, b12 – b22, d3) **Step 8:** Recursively call Strassen (n/2, a22, b21 – b11, d4) **Step 9:** Recursively call Strassen (n/2, a11 + a12, b22, d5) **Step 10:** Recursively call Strassen (n/2, a21 – a11, b11 + b22, d6) **Step 11:** Recursively call Strassen (n/2, a12 – a22, b21 + b22, d7) **Step 12:** Calculate matrix C elements **Step 13:** Stop
Complexity	Addition and Subtraction of two matrices takes O(N^2) time. So time complexity can be written as: T(N) = 7T(N/2) + O(N^2) From Master's Theorem, time complexity of above method is O(N^Log7) which is approximately O(N^2.8074)

Program

main.c Run

```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("Shivam Tawari \nSem-3 \nA-49\n");

  printf("Enter the 4 elements of first matrix: \n");
  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("\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	Output
	`gcc -o /tmp/b4xOw5pO32.o /tmp/b4xOw5pO32.c -lm`
	`/tmp/b4xOw5pO32.o`
	`Shivam Tawari`
	`Sem-3`
	`A-49`
	`Enter the 4 elements of first matrix:`
	`3`
	`2`
	`6`
	`5`
	`Enter the 4 elements of second matrix: 1`
	`5`
	`2`
	`7`
	`The first matrix is`
	
	`3   2`
	`6   5`
	`The second matrix is`
	
	`1   5`
	`2   7`
	`After multiplication using Strassen's algorithm`
	
	`7   29`
	`16   65`