Title:Implementation of Strassen's Matrix multiplication.

Theory/Description:

using Strassen's Matrix multiplication algorithm, the time consumption can 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=[IKJL]Z=[IJKL] X=[ACBD]X=[ABCD] and Y=[EGFH]Y=[EFGH]
Using Strassen's Algorithm compute the following −

$$M1:=(A+C)\times(E+F)$$
$$M2:=(B+D)\times(G+H)$$
$$M3:=(A-D)\times(E+H)$$
$$M4:=A\times(F-H)$$
$$M5:=(C+D)\times(E)$$
$$M6:=(A+B)\times(H)$$
$$M7:=D\times(G-E)$$

Then,

$$I:=M2+M3-M6-M7$$
$$J:=M4+M6$$
$$K:=M5+M7$$
$$L:=M1-M3-M4-M5$$

Analysis:

T(n)={c ..... fn=1

7xT(n2)+dxn2otherwise, where *c* and *d* are constants
Using this recurrence relation, we get T(n)=O(nlog7)T(n)=O(nlog7)
Hence, the complexity of Strassen's matrix multiplication algorithm is O(nlog7)O(nlog7).

```
C code of two 2 by 2 matrix multiplication using Strassen's algorithm
*/
#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("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("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;
}
```

VIVA-VOCE QUESTIONS:

1. Is there any optimum solution for Matrixmultiplication?
   Ans: Yes. Divide and conquer method suggests Strassen's matrix multiplication method to be used. If we follow this method, the time complexity is O(n*n*n……..*2.81) times rather O(n*n*n*……*3) times.