|  |  |
| --- | --- |
| **Name** | Parth Gandhi |
| **UID No.** | 2021300033 |
| **Class & Division** | S.E. COMPS A (BATCH B) |
| **Experiment No.** | 3 |

**Aim:** Experiment on performing Strassen’s matrix multiplication.

**Theory:**

Strassen's algorithm is an efficient way to multiply two matrices. It is based on a divide-and-conquer approach and was developed by Volker Strassen in 1969.

The basic idea behind the Strassen's algorithm is to divide the two matrices to be multiplied into smaller sub-matrices, and then recursively compute the product of these sub-matrices. This leads to a reduction in the number of multiplications required to compute the product, and hence improves the overall efficiency of the multiplication.

**Algorithm:**

1. Given two matrices A and B, we need to multiply them to obtain the matrix C.
2. Check if the matrices are square and have the same dimensions. If not, pad the matrices with zeroes to make them square and of equal dimensions.
3. Divide both matrices A and B into four sub-matrices each, such that each sub-matrix has n/2 rows and n/2 columns. Here n is the dimension of the square matrices.
4. Compute seven matrix products recursively using the sub-matrices obtained in step 3. These matrix products are:
   1. P1 = A11 x (B12 - B22)
   2. P2 = (A11 + A12) x B22
   3. P3 = (A21 + A22) x B11
   4. P4 = A22 x (B21 - B11)
   5. P5 = (A11 + A22) x (B11 + B22)
   6. P6 = (A12 - A22) x (B21 + B22)
   7. P7 = (A11 - A21) x (B11 + B12)
5. Here A11, A12, A21, and A22 are the four sub-matrices of matrix A, and B11, B12, B21, and B22 are the four sub-matrices of matrix B.
6. Compute the four sub-matrices of matrix C using the products obtained in step 4. These sub-matrices are:
   1. C11 = P5 + P4 - P2 + P6
   2. C12 = P1 + P2
   3. C21 = P3 + P4
   4. C22 = P5 + P1 - P3 - P7
7. Here C11, C12, C21, and C22 are the four sub-matrices of matrix C.
8. Combine the four sub-matrices obtained in step 5 to form the final matrix C.
9. Return the matrix C as the result of matrix multiplication of A and B.

**Code:**

#include <bits/stdc++.h>

using namespace std;

int main(){

    Int a[2][2],b[2][2],c[2][2];

    cout<<"Enter the elements of 2x2 Matrix A:\n";

    for(int i=0;i<2;i++){

        for(int j=0;j<2;j++)

        cin>>a[i][j];

    }

    cout<<"Enter the elements of 2x2 Matrix B:\n";

    for(int i=0;i<2;i++){

        for(int j=0;j<2;j++)

        cin>>b[i][j];

    }

    int s1=b[0][1]-b[1][1];

    int s2=a[0][0]+a[0][1];

    int s3=a[1][0]+a[1][1];

    int s4=b[1][0]-b[0][0];

    int s5=a[0][0]+a[1][1];

    int s6=b[0][0]+b[1][1];

    int s7=a[0][1]-a[1][1];

    int s8=b[1][0]+b[1][1];

    int s9=a[0][0]-a[1][0];

    int s10=b[0][0]+b[0][1];

    int p1=a[0][0]\*s1;

    int p2=b[1][1]\*s2;

    int p3=b[0][0]\*s3;

    int p4=a[1][1]\*s4;

    int p5=s5\*s6;

    int p6=s7\*s8;

    int p7=s9\*s10;

    c[0][0]=p5+p4-p2+p6;

    c[0][1]=p1+p2;

    c[1][0]=p3+p4;

    c[1][1]=p5+p1-p3-p7;

    cout<<"\nProduct of A and B is:\n";

    for(int i=0;i<2;i++){

        for(int j=0;j<2;j++)

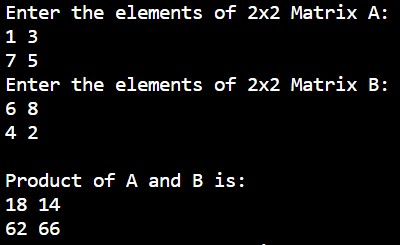
        cout<<c[i][j]<<" ";

        cout<<"\n";

    }

}

**Output:**



**Conclusion:** Successfully wrote a program to implement Strassen matrix mutliplication.