|  |  |
| --- | --- |
| **Name** | Rupali Sawale |
| **UID no.** | 2021700056 |
| **Experiment No.** | 3 |

|  |  |
| --- | --- |
| **AIM:** | Implement Divide and Conquer technique. |
| **Program** | |
| **PROBLEM STATEMENT :** | Implement Strassen’s Matrix Multiplication algorithm and compare it with standard matrix multiplication. |
| **ALGORITHM/**  **THEORY:** | // 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;} |
| **Result:**  OUTPUT 1:    Verification of algorithm:    OUTPUT2:    Verification of algorithm: | |
| **CONCLUSION:** | We used Strassen’s Matrix Multiplication and compared with Standard Matrix Multiplication logic. On comparison we found out that Strassen’s is better than standard method for multiplication of square matrices. |