|  |  |
| --- | --- |
| **NAME:** | Shubham Solanki |
| **UID:** | 2022301015 |
| **SUBJECT** | Design and Analysis of Algorithms |
| **EXPERIMENT NO:** | 3 |
| **AIM:** | Strassen’s Matrix Multiplication |
| **Algorithm:** | **Strassen’s Matrix Multiplication Algorithm**  **Step 1: Start**  **Step 2: Take 2 matrices as input from user say A and B**  **Step 3: Divide A and B into 10 matrices of n/2 size**  S[0] = B[0][1] - B[1][1];  S[1] = A[0][0] + A[0][1];  S[2] = A[1][0] + A[1][1];  S[3] = B[1][0] - B[0][0];  S[4] = A[0][0] + A[1][1];  S[5] = B[0][0] + B[1][1];  S[6] = A[0][1] - A[1][1];  S[7] = B[1][0] + B[1][1];  S[8] = A[0][0] - A[1][0];  S[9] = B[0][0] + B[0][1];  **Step 4: Compute p1 to p7**  P[0] = A[0][0] \* S[0];  P[1] = B[1][1] \* S[1];  P[2] = B[0][0] \* S[2];  P[3] = A[1][1] \* S[3];  P[4] = S[5] \* S[4];  P[5] = S[6] \* S[7];  P[6] = S[8] \* S[9];  **Step 5: computer the resultant matrix c:**  C[0][0] = P[4] + P[3] - P[1] + P[5];  C[0][1] = P[0] + P[1];  C[1][0] = P[2] + P[3];  C[1][1] = P[4] + P[0] - P[2] - P[6];  **Step 6: display the matrix C**  **Step 7: End** |
| **Code:** | **2 x 2 Matrix**  #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("\nEnter the 4 elements of first matrix: ");    for(i = 0;i < 2; i++)        for(j = 0;j < 2; j++)             scanf("%d", &a[i][j]);    printf("\nEnter the 4 elements of second matrix: ");    for(i = 0; i < 2; i++)        for(j = 0;j < 2; j++)             scanf("%d", &b[i][j]);    printf("\n\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("\n\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("\n\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]);     }     printf("\n\n");     return 0;  } |
| **Graphs and Observation:** | **2 x 2 Matrix** |
| **Conclusion:** | Thus, after performing this experiment I understood that Strassen’s matrix multiplication is very efficient as it improves the run time a lot when multiplying matrices than traditional matrix multiplication. Strassen Matrix multiplication is very easy to implement but it requires a lot of space as we need to store multiple arrays. |