

Name	Stephen David Vaz
UID no.	2021700070
Experiment No.	3

AIM:	Strassen's Matrix Multiplication
PROBLEM STATEMENT :	Implement Strassen's Matrix Multiplication for a matrix of order 2
ALGORITHM/THEORY:	<p>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 for large matrices, with a better asymptotic complexity, although the naive algorithm is often better for smaller matrices. The Strassen algorithm is slower than the fastest known algorithms for extremely large matrices, but such galactic algorithms are not useful in practice, as they are much slower for matrices of practical size. For small matrices even faster algorithms exist.</p> <p>Strassen's algorithm works for any ring, such as plus/multiply, but not all semirings, such as min-plus or boolean algebra, where the naive algorithm still works, and so called combinatorial matrix multiplication.</p> <ol style="list-style-type: none"> 1. Divide a matrix of the order of 2×2 recursively until we get the matrix of order 2×2. 2. To carry out the multiplication of the 2×2 matrix, use the previous set of formulas. 3. Subtraction is also performed within these eight multiplications and four additions. 4. To find the final product or final matrix combine the result of two matrices, use the below formulae <p> $D1 = (a11 + a22) * (b11 + b22)$ $D2 = (a21 + a22) * b11$ $D3 = (b12 - b22) * a11$ $D4 = (b21 - b11) * a22$ $D5 = (a11 + a12) * b22$ $D6 = (a21 - a11) * (b11 + b12)$ $D7 = (a12 - a22) * (b21 + b22)$ </p>

	$C00 = d1 + d4 - d5 + d7$ $C01 = d3 + d5$ $C10 = d2 + d4$ $C11 = d1 + d3 - d2 - d6$ <p>Here, C00, C01, C10, and C11 are the elements of the 2*2 matrix.</p>
PROGRAM:	<pre> #include <stdio.h> #include <time.h> void printarr(int a[][2]) { for (int i = 0; i < 2; i++) { for (int j = 0; j < 2; j++) { printf("%d\t", a[i][j]); } printf("\n\n"); } } void userInput(int a[][2]) { for (int i = 0; i < 2; i++) { for (int j = 0; j < 2; j++) { int input; scanf("%d", &input); a[i][j] = input; } } } int main() { int mat1[2][2], mat2[2][2], matres[2][2]; int m1, m2, m3, m4, m5, m6, m7; printf("Matrix 1:-\n"); </pre>

```

userInput(mat1);
printf("Matrix 2:-\n");
userInput(mat2);
printf("\nMatrix 1:-\n");
printarr(mat1);
printf("Matrix 2:-\n");
printarr(mat2);

//strassen multiplication algorithm
clock_t start, end;
double cpu_time_used;
start = clock();
m1 = (mat1[0][0] + mat1[1][1]) * (mat2[0][0] + mat2[1][1]);
m2 = (mat1[1][0] + mat1[1][1]) * mat2[0][0];
m3 = mat1[0][0] * (mat2[0][1] - mat2[1][1]);
m4 = mat1[1][1] * (mat2[1][0] - mat2[0][0]);
m5 = (mat1[0][0] + mat1[0][1]) * mat2[1][1];
m6 = (mat1[1][0] - mat1[0][0]) * (mat2[0][0] + mat2[0][1]);
m7 = (mat1[0][1] - mat1[1][1]) * (mat2[1][0] + mat2[1][1]);

matres[0][0] = m1 + m4 - m5 + m7;
matres[0][1] = m3 + m5;
matres[1][0] = m2 + m4;
matres[1][1] = m1 - m2 + m3 + m6;
end = clock();
cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
printf("Matrix1 X Matrix2:-\n");
printf("Strassen(time): %f\n", cpu_time_used);
printarr(matres);

//itterative multiplication algorithm
int res[2][2] = {0};
start = clock();
for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 2; j++)
    {
        for (int k = 0; k < 2; k++)
        {
            res[i][j] += mat1[i][k] * mat2[k][j];
        }
    }
}

```

```

    }
}

end = clock();
cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
printf("Matrix1 X Matrix2:-\n");
printf("Iterative(time): %f\n", cpu_time_used);
printarr(res);
return 0;
}

```

RESULT:

● * Executing task: /usr/bin/clang /Users/stephen03/Dev/repos/stepDAA/exp3/mat.c -o ../exc
s/mat && ../exc/s/mat

```

Matrix 1:-
123 234 345 456
Matrix 2:-
567 678 789 891

```

```

Matrix 1:-
123    234

345    456

```

```

Matrix 2:-
567    678

789    891

```

```

Matrix1 X Matrix2:-
Strassen(time): 0.000010
254367    291888

555399    640206

```

```

Matrix1 X Matrix2:-
Iterative(time): 0.000002
254367    291888

555399    640206

```

* Terminal will be reused by tasks, press any key to close it.

```

● * Executing task: /usr/bin/clang /Users/stephen03/Dev/repos/stepDAA/exp3/mat.c -o ../exc
s/mat && ../excs/mat

Matrix 1:-
1 2 3 4
Matrix 2:-
4 3 2 1

Matrix 1:-
1 2
3 4
Matrix 2:-
4 3
2 1
Matrix1 X Matrix2:-
Strassen(time): 0.000006
8 5
20 13

Matrix1 X Matrix2:-
Iterative(time): 0.000002
8 5
20 13

* Terminal will be reused by tasks, press any key to close it.

```

CONCLUSION:

Successfully understood matrix multiplication using strassen's matrix multiplication algorithm. Also, compared time required by both strassen's as well as iterative multiplication algorithm.