Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058, India
(Autonomous College Affiliated to University of Mumbai)

Experiment No.	3
Aim	To understand and implement Strassen's Matrix Multiplication.
Name	Pratiksha Pravin Patil
UID No.	2022301016
Class & Division	Comp(A)

Theory:

Strassen's matrix multiplication algorithm is an efficient algorithm for multiplying two matrices. It was developed by Volker Strassen in 1969 and is based on the divide-and-conquer approach.

The traditional matrix multiplication algorithm has a time complexity of O(n^3) for multiplying two n x n matrices. Strassen's algorithm reduces this time complexity to O(n^log2(7)), which is approximately O(n^2.81).

The algorithm works by dividing the matrices into smaller submatrices and recursively multiplying them. The key insight of Strassen's algorithm is that instead of computing the eight individual products needed for matrix multiplication, we can compute just seven products by cleverly combining the submatrices.

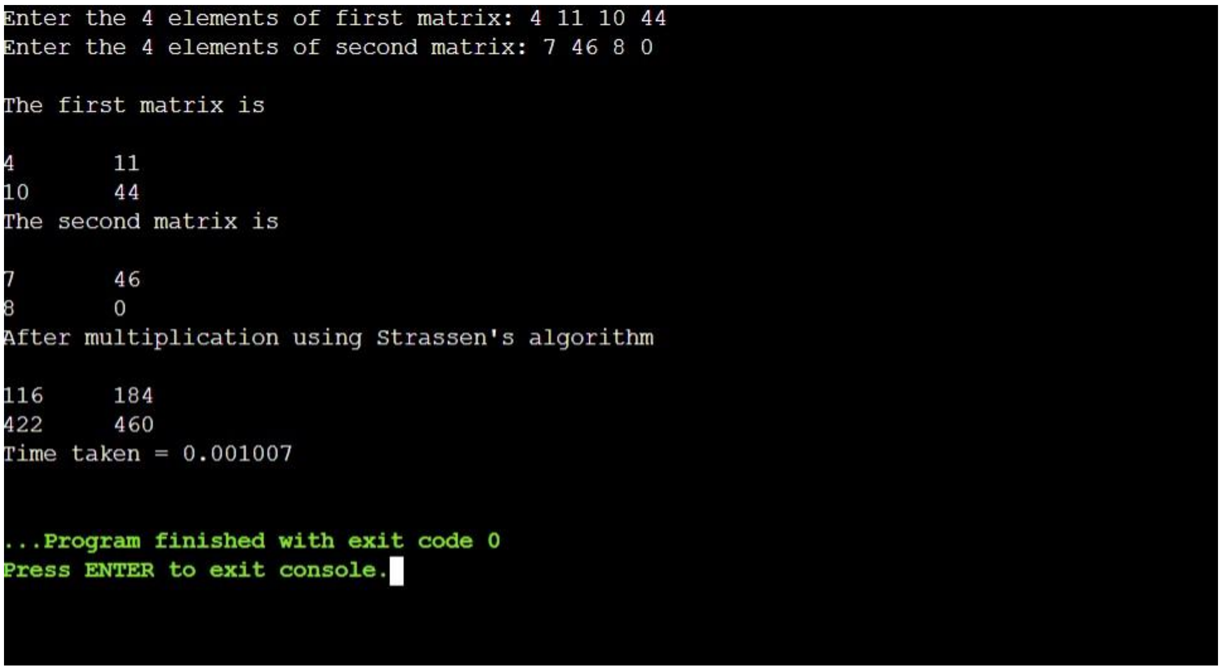
The algorithm is as follows:

1. Divide the input matrices A and B into four equally sized submatrices: A11, A12, A21, A22 and B11, B12, B21, B22.
2. Compute seven matrix products recursively: P1 = A11 x (B12 - B22) P2 = (A11 + A12) x B22 P3 = (A21 + A22) x B11 P4 = A22 x (B21 - B11) P5 = (A11 + A22) x (B11 + B22) P6 = (A12 - A22) x (B21 + B22) P7 = (A11 - A21) x (B11 + B12)
3. Compute the four submatrices of the resulting matrix C: C11 = P5 + P4 - P2 + P6 C12 = P1 + P2 C21 = P3 + P4 C22 = P5 + P1 - P3 - P7
4. Combine the four submatrices to form the final output matrix C.

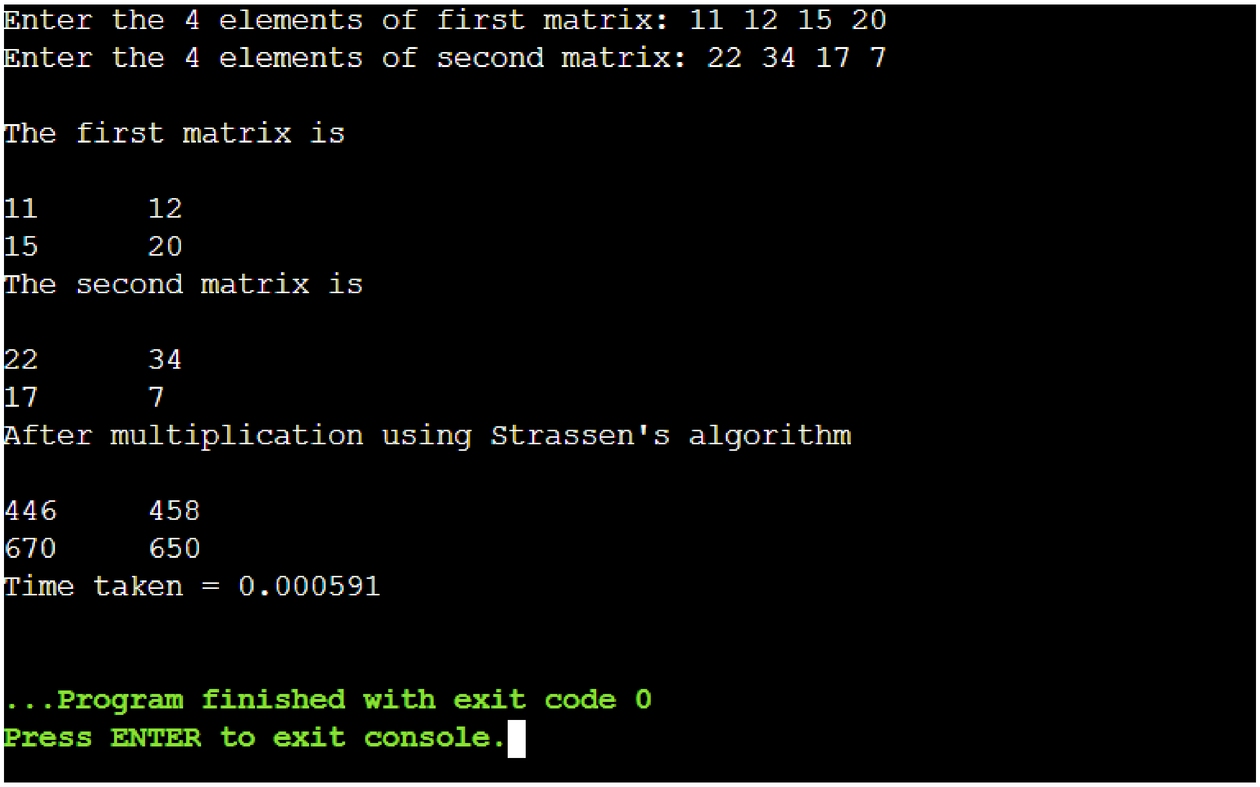
Program:

```
#include<stdio.h>
#include<time.h>
int main(){

  int a[2][2], b[2][2], c[2][2], i, j;
  int m1, m2, m3, m4 , m5, m6, m7;
   clock_t start,end;
  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]);
   }
   printf("\n");
end=clock();
printf("Time taken = %lf\n",(double)(end-start)/CLOCKS_PER_SEC);

   return 0;
}
```

Output:



```
Enter the 4 elements of first matrix: 4 11 10 44
Enter the 4 elements of second matrix: 7 46 8 0

The first matrix is

4       11
10      44
The second matrix is

7       46
8       0
After multiplication using Strassen's algorithm

116     184
422     460
Time taken = 0.001007


...Program finished with exit code 0
Press ENTER to exit console.
```



```
Enter the 4 elements of first matrix: 11 12 15 20
Enter the 4 elements of second matrix: 22 34 17 7

The first matrix is

11      12
15      20
The second matrix is

22      34
17      7
After multiplication using Strassen's algorithm

446     458
670     650
Time taken = 0.000591


...Program finished with exit code 0
Press ENTER to exit console.
```

Conclusion: In conclusion, Strassen's matrix multiplication algorithm is an efficient algorithm for multiplying two matrices by using the divide-and-conquer approach. It reduces the time complexity of matrix multiplication from O(n^3) to O(n^log2(7)), which is approximately O(n^2.81). The algorithm works by dividing the matrices into smaller submatrices and recursively multiplying them. By cleverly combining the submatrices, the algorithm only requires seven products instead of the eight products needed for traditional matrix multiplication. Overall, Strassen's algorithm is a useful tool for multiplying large matrices efficiently.