DAA 3

Mayank Rohilla

October 2023

1 strassen's matrix multiplication

Strassen's matrix multiplication is an algorithm for matrix multiplication that uses a divide-and-conquer approach to reduce the number of multiplications needed to calculate the product of two matrices.The algorithm divides each matrix into submatrices and recursively multiplies them using a set of seven multiplications and some additional additions and subtractions which is faster than the standard matrix multiplication method.. However, Strassen's algorithm is not always preferred for practical applications because the constants used in the algorithm are high and the submatrices in recursion take extra space. Practical implementations of Strassen's algorithm switch to standard methods of matrix multiplication for small enough submatrices, for which those algorithms are more efficient

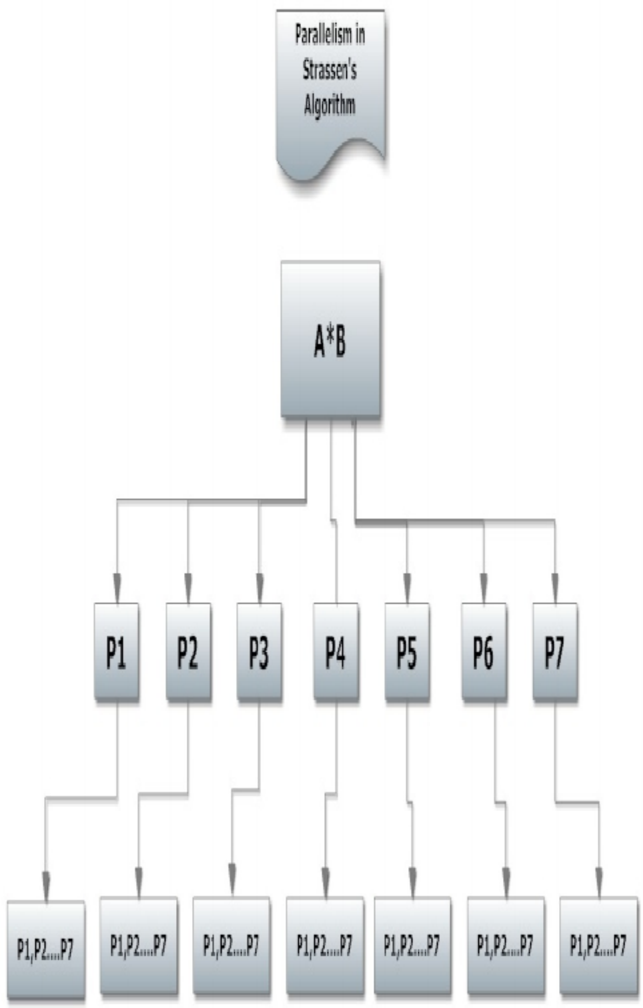
1.1 Algorithm

Strassen's matrix multiplication algorithm is a divide-and-conquer approach to solving matrix multiplication problems.The algorithm is named after Volker Strassen, who introduced it in 1969. The algorithm is faster than the standard matrix multiplication algorithm for large matrices.which is faster than the standard matrix multiplication method. The algorithm works by dividing the matrices A and B into smaller submatrices of size n/2 x n/2, computing seven products recursively using the submatrices of A and B, computing the submatrices of the result matrix C, and combining the submatrices into the final result matrix . The seven products are computed using scalar addition and subtraction of the submatrices . The algorithm is not always preferred for practical applications because the constants used in the algorithm are high and the submatrices in recursion take extra space . Practical implementations of Strassen's algorithm switch to standard methods of matrix multiplication for small enough submatrices, for which those algorithms are more efficient

To implement Strassen's matrix multiplication algorithm, we can follow the steps below:

- Divide the matrices A and B into smaller submatrices of size n/2 x n/2.
- Compute seven products recursively using the submatrices of A and B.
- Compute the submatrices of the result matrix C.
- Combine the submatrices of C into the final result matrix.

1.2 Flowchart



1.3 Code

```c
/*
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;
}
```

1.4 Output

```
Enter the 4 elements of first matrix: 1
2
3
4
Enter the 4 elements of second matrix: 5
6
7
8

The first matrix is

1       2
3       4
The second matrix is

5       6
7       8
After multiplication using Strassen's algorithm

19      22
43      50
Press any key to continue . . .
```