# Strassen's Matrix Multiplication

Dr. Anand M
Id: 102763
Assistant Professor
Department of Computer Science and Engineering
SRM Institute of Science and Technology

# Basic Matrix Multiplication

Suppose we want to multiply two matrices of size $N$ x $N$: for example $A$ x $B$ = $C$.

$$\begin{vmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{vmatrix} = \begin{vmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{vmatrix} \begin{vmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{vmatrix}$$

$$\mathbf{C_{11} = a_{11}b_{11} + a_{12}b_{21}}$$

$$\mathbf{C_{12} = a_{11}b_{12} + a_{12}b_{22}}$$

$$\mathbf{C_{21} = a_{21}b_{11} + a_{22}b_{21}}$$

$$\mathbf{C_{22} = a_{21}b_{12} + a_{22}b_{22}}$$

2x2 matrix multiplication can be accomplished in 8 multiplication.$\mathbf{(2^{\log_2 8} = 2^3)}$

# Basic Matrix Multiplication

```
void matrix_mult (){
  for (i = 1; i <= N; i++) {
        for (j = 1; j <= N; j++) {
        compute C_{i,j};
                          }
  }}
```

algorithm

Time analysis

$$C_{i,j} = \sum_{k=1}^{N} a_{i,k} b_{k,j}$$

$$\text{Thus } T(N) = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} c = cN^3 = O(N^3)$$

# Strassens's Matrix Multiplication

- Strassen showed that 2x2 matrix multiplication can be accomplished in 7 multiplication and 18 additions or subtractions. .$(2^{\log_2 7} = 2^{2.807})$

- This reduce can be done by Divide and Conquer Approach.

# Divide-and-Conquer

- Divide-and conquer is a general algorithm design paradigm:
  - Divide: divide the input data $\boldsymbol{S}$ in two or more disjoint subsets $\boldsymbol{S}_1$, $\boldsymbol{S}_2$, …
  - Recur: solve the subproblems recursively
  - Conquer: combine the solutions for $\boldsymbol{S}_1$, $\boldsymbol{S}_2$, …, into a solution for $\boldsymbol{S}$
- The base case for the recursion are subproblems of constant size
- Analysis can be done using **recurrence equations**

# Divide and Conquer Matrix Multiply

$$A \times B = R$$

$$\begin{bmatrix} A_0 & A_1 \\ A_2 & A_3 \end{bmatrix} \times \begin{bmatrix} B_0 & B_1 \\ B_2 & B_3 \end{bmatrix} = \begin{bmatrix} A_0 \times B_0 + A_1 \times B_2 & A_0 \times B_1 + A_1 \times B_3 \\ A_2 \times B_0 + A_3 \times B_2 & A_2 \times B_1 + A_3 \times B_3 \end{bmatrix}$$

- Divide matrices into sub-matrices: $A_0$ , $A_1$, $A_2$ etc
- Use blocked matrix multiply equations
- Recursively multiply sub-matrices

# Divide and Conquer Matrix Multiply

$$A \times B = R$$

$$\boxed{a_0} \times \boxed{b_0} = \boxed{a_0 \times b_0}$$

- Terminate recursion with a simple base case

# Strassens's Matrix Multiplication

$$\begin{vmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{vmatrix} = \begin{vmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{vmatrix} \begin{vmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{vmatrix}$$

$P_1 = (A_{11} + A_{22})(B_{11} + B_{22})$
$P_2 = (A_{21} + A_{22}) * B_{11}$
$P_3 = A_{11} * (B_{12} - B_{22})$
$P_4 = A_{22} * (B_{21} - B_{11})$
$P_5 = (A_{11} + A_{12}) * B_{22}$
$P_6 = (A_{21} - A_{11}) * (B_{11} + B_{12})$
$P_7 = (A_{12} - A_{22}) * (B_{21} + B_{22})$

$C_{11} = P_1 + P_4 - P_5 + P_7$
$C_{12} = P_3 + P_5$
$C_{21} = P_2 + P_4$
$C_{22} = P_1 + P_3 - P_2 + P_6$

# Comparison

$$
\begin{aligned}
C_{11} &= P_1 + P_4 - P_5 + P_7 \\
&= (A_{11} + A_{22})(B_{11} + B_{22}) + A_{22} * (B_{21} - B_{11}) - (A_{11} + A_{12}) * B_{22} + \\
&\quad (A_{12} - A_{22}) * (B_{21} + B_{22}) \\
&= A_{11} B_{11} + A_{11} B_{22} + A_{22} B_{11} + A_{22} B_{22} + A_{22} B_{21} - A_{22} B_{11} - \\
&\quad A_{11} B_{22} - A_{12} B_{22} + A_{12} B_{21} + A_{12} B_{22} - A_{22} B_{21} - A_{22} B_{22} \\
&= A_{11} B_{11} + A_{12} B_{21}
\end{aligned}
$$

# Strassen Algorithm

```
void matmul(int *A, int *B, int *R, int n) {
  if (n == 1) {
        (*R) += (*A) * (*B);
  } else {
        matmul(A, B, R, n/4);
        matmul(A, B+(n/4), R+(n/4), n/4);
        matmul(A+2*(n/4), B, R+2*(n/4), n/4);
        matmul(A+2*(n/4), B+(n/4), R+3*(n/4), n/4);
        matmul(A+(n/4), B+2*(n/4), R, n/4);
        matmul(A+(n/4), B+3*(n/4), R+(n/4), n/4);
        matmul(A+3*(n/4), B+2*(n/4), R+2*(n/4), n/4);
        matmul(A+3*(n/4), B+3*(n/4), R+3*(n/4), n/4);
  }
```

Divide matrices in sub-matrices and recursively multiply sub-matrices

# Time Analysis

$$T(1) = 1 \qquad (\text{assume } N = 2^k)$$

$$T(N) = 7T(N/2)$$

$$T(N) = 7^k T(N/2^k) = 7^k$$

$$T(N) = 7^{\log N} = N^{\log 7} = N^{2.81}$$

# Assignment Work

1. Verify the formulas of Strassen's algorithm for multiplying 2 matrices.
2. Apply Strassen's algorithm to compute.

$$\begin{bmatrix} 1 & 0 & 2 & 1 \\ 4 & 1 & 1 & 0 \\ 0 & 1 & 3 & 0 \\ 5 & 0 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 & 1 \\ 2 & 1 & 0 & 4 \\ 2 & 0 & 1 & 1 \\ 1 & 3 & 5 & 0 \end{bmatrix}$$