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 $A \times B = C$.

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

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

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

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

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

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

Basic Matrix Multiplication

```
void matrix_mult (){
  for (i = 1; i <= N; i++) {
    for (j = 1; j <= N; j++) {
      compute C<sub>i,j</sub>;
      }
}}
```

algorithm

Time analysis

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

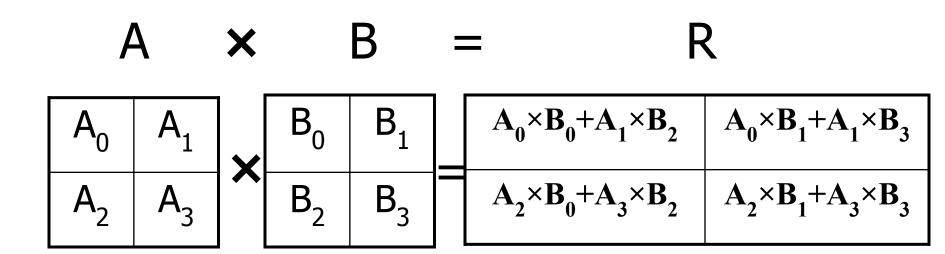
• Strassen showed that $2x^2$ matrix multiplication can be accomplished in 7 multiplication and 18 additions or subtractions. $(2^{\log_2 7} = 2^{2.807})$

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

Divide-and-Conquer

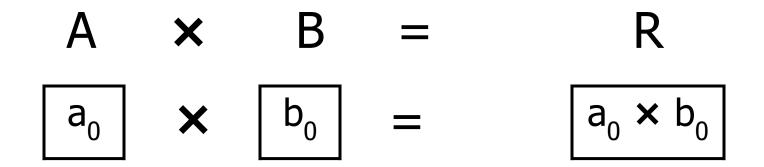
- Divide-and conquer is a general algorithm design paradigm:
 - Divide: divide the input data S in two or more disjoint subsets $S_1, S_2, ...$
 - Recur: solve the subproblems recursively
 - Conquer: combine the solutions for $S_1, S_2, ...,$ into a solution for S
- The base case for the recursion are subproblems of constant size
- Analysis can be done using recurrence equations

Divide and Conquer Matrix Multiply



- •Divide matrices into sub-matrices: A₀, A₁, A₂ etc
- •Use blocked matrix multiply equations
- •Recursively multiply sub-matrices

Divide and Conquer Matrix Multiply



• Terminate recursion with a simple base case

Strassens's Matrix Multiplication

$$\left| \begin{array}{cc|c} C_{11} & C_{12} \\ C_{21} & C_{22} \end{array} \right| = \left| \begin{array}{cc|c} A_{11} & A_{12} \\ A_{21} & A_{22} \end{array} \right| \left| \begin{array}{cc|c} B_{11} & B_{12} \\ B_{21} & B_{22} \end{array} \right|$$

$$\begin{split} \mathbf{P}_1 &= (\mathbf{A}_{11} + \mathbf{A}_{22})(\mathbf{B}_{11} + \mathbf{B}_{22}) \\ \mathbf{P}_2 &= (\mathbf{A}_{21} + \mathbf{A}_{22}) * \mathbf{B}_{11} \\ \mathbf{P}_3 &= \mathbf{A}_{11} * (\mathbf{B}_{12} - \mathbf{B}_{22}) \\ \mathbf{P}_4 &= \mathbf{A}_{22} * (\mathbf{B}_{21} - \mathbf{B}_{11}) \\ \mathbf{P}_5 &= (\mathbf{A}_{11} + \mathbf{A}_{12}) * \mathbf{B}_{22} \\ \mathbf{P}_6 &= (\mathbf{A}_{21} - \mathbf{A}_{11}) * (\mathbf{B}_{11} + \mathbf{B}_{12}) \\ \mathbf{P}_7 &= (\mathbf{A}_{12} - \mathbf{A}_{22}) * (\mathbf{B}_{21} + \mathbf{B}_{22}) \end{split}$$

Comparison

$$\begin{split} \mathbf{C}_{11} &= \mathbf{P}_1 + \mathbf{P}_4 - \mathbf{P}_5 + \mathbf{P}_7 \\ &= (\mathbf{A}_{11} + \mathbf{A}_{22})(\mathbf{B}_{11} + \mathbf{B}_{22}) + \mathbf{A}_{22} * (\mathbf{B}_{21} - \mathbf{B}_{11}) - (\mathbf{A}_{11} + \mathbf{A}_{12}) * \mathbf{B}_{22} + \\ &\quad (\mathbf{A}_{12} - \mathbf{A}_{22}) * (\mathbf{B}_{21} + \mathbf{B}_{22}) \\ &= \mathbf{A}_{11} \mathbf{B}_{11} + \mathbf{A}_{11} \mathbf{B}_{22} + \mathbf{A}_{22} \mathbf{B}_{11} + \mathbf{A}_{22} \mathbf{B}_{22} + \mathbf{A}_{22} \mathbf{B}_{21} - \mathbf{A}_{22} \mathbf{B}_{11} - \\ &\quad \mathbf{A}_{11} \mathbf{B}_{22} - \mathbf{A}_{12} \mathbf{B}_{22} + \mathbf{A}_{12} \mathbf{B}_{21} + \mathbf{A}_{12} \mathbf{B}_{22} - \mathbf{A}_{22} \mathbf{B}_{21} - \mathbf{A}_{22} \mathbf{B}_{22} \\ &= \mathbf{A}_{11} \mathbf{B}_{11} + \mathbf{A}_{12} \mathbf{B}_{21} \end{split}$$

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

$$T(1) = 1$$
 (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

- 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}
```