UNIT II - DIVIDE-AND-CONQUER

Divide and Conquer Methodology – Binary Search – Merge Sort – Quick Sort – Heap Sort – Multiplication of Large Integers – **Strassen's Matrix Multiplication**

Strassens's Matrix Multiplication

- published by V. Strassen in 1969
- The principal insight of the algorithm lies in the discovery that we can find the product C of two 2 × 2 matrices A and B with just seven multiplications as opposed to the eight required by the brute-force algorithm
- This is accomplished by using the formulas

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.

- **Brute-force algorithm**

$$\begin{pmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} * \begin{pmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{pmatrix}$$

$$= \begin{pmatrix} a_{00} * b_{00} + a_{01} * b_{10} & a_{00} * b_{01} + a_{01} * b_{11} \\ a_{10} * b_{00} + a_{11} * b_{10} & a_{10} * b_{01} + a_{11} * b_{11} \end{pmatrix}$$

8 multiplications

4 additions

Efficiency class in general: $\Theta(n^3)$

Basic Matrix Multiplication

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

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

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

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

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

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

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_{i,j};
                                }
  }}
```

algorithm

Time analysis

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

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

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 $\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots$
  - Recur: solve the subproblems recursively
  - Conquer: combine the solutions for $\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots$, 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

- **Strassen's algorithm for two 2x2 matrices (1969):**

$$\begin{pmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} * \begin{pmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{pmatrix}$$

$$= \begin{pmatrix} m_1 + m_4 - m_5 + m_7 & m_3 + m_5 \\ m_2 + m_4 & m_1 + m_3 - m_2 + m_6 \end{pmatrix}$$

- $m_1 = (a_{00} + a_{11}) * (b_{00} + b_{11})$
- $m_2 = (a_{10} + a_{11}) * b_{00}$
- $m_3 = a_{00} * (b_{01} - b_{11})$
- $m_4 = a_{11} * (b_{10} - b_{00})$
- $m_5 = (a_{00} + a_{01}) * b_{11}$
- $m_6 = (a_{10} - a_{00}) * (b_{00} + b_{01})$
- $m_7 = (a_{01} - a_{11}) * (b_{10} + b_{11})$

7 multiplications

18 additions

Strassen observed [1969] that the product of two matrices can be computed in general as follows:

$$\begin{pmatrix} C_{00} & C_{01} \\ C_{10} & C_{11} \end{pmatrix} = \begin{pmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{pmatrix} * \begin{pmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{pmatrix}$$

$$= \begin{pmatrix} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ M_2 + M_4 & M_1 + M_3 - M_2 + M_6 \end{pmatrix}$$

Formulas for Strassen's Algorithm

$M_1 = (A_{00} + A_{11}) * (B_{00} + B_{11})$
$M_2 = (A_{10} + A_{11}) * B_{00}$
$M_3 = A_{00} * (B_{01} - B_{11})$
$M_4 = A_{11} * (B_{10} - B_{00})$
$M_5 = (A_{00} + A_{01}) * B_{11}$
$M_6 = (A_{10} - A_{00}) * (B_{00} + B_{01})$
$M_7 = (A_{01} - A_{11}) * (B_{10} + B_{11})$

$S_1 = (A_{00} + A_{11})$
$S2 = (B_{00} + B_{11})$
$S3 = (A_{10} + A_{11})$
$S4 = (B_{01} - B_{11})$
$S5 = (B_{10} - B_{00})$
$S6 = (A_{00} + A_{01})$
$S7 = (A_{10} - A_{00})$
$S8 = (B_{00} + B_{01})$
$S9 = (A_{01} - A_{11})$
$S10 = (B_{10} + B_{11})$

$$\begin{pmatrix} M1 + M4 - M5 + M7 & M3 + M5 \\ M2 + M4 & M1 + M3 - M2 + M6 \end{pmatrix}$$

7 multiplications

18 additions

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

Analysis of Strassen's Algorithm

If *n* is not a power of 2, matrices can be padded with zeros.

Number of multiplications: $M(n) = 7M(n/2)$, $M(1) = 1$

Solution: $M(n) = 7^{\log_2 n} = n^{\log_2 7} \approx n^{2.807}$ vs. n^3 of brute-force alg.

Algorithms with better asymptotic efficiency are known but they are even more complex and not used in practice.

Time Analysis

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

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

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

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

MULTIPLIES A and B using Strassen's O (n^2.81) method

A = [1 3] B = [6 8]
[7 5] [4 2]

Step 1: Split A and B into half-sized matrices of size 1x1 (scalars).

a11 = 1

a12 = 3

a21 = 7

a22 = 5

b11 = 6

b12 = 8

b21 = 4

s1 = b12 - b22 = 8 - 2 = 6

s2 = a11 + a12 = 1 + 3 = 4

s3 = a21 + a22 = 7 +5 = 12

s4 = b21 - b11 = 4 – 6 = -2

s5 = a11 + a22 = 1 + 5 = 6

s6 = b11 + b22 = 6 + 2 = 8

s7 = a12 - a22 = 3 – 5 = -2

s8 = b21 + b22 = 4 + 2 = 6

s9 = a11 - a21 = 1 – 7 = -6

s10 = b11 + b12 = 6 + 8 =14

p1 = a11 * s1 = 1 * 6 = 6

p2 = s2 * b22 = 4 * 2 = 8

p3 = s3 * b11 = 12 * 6 = 72

p4 = a22 * s4 = 5 * -2 = -10

p5 = s5 * s6 = 6 * 8 = 48

p6 = s7 * s8 = -2 * 6 = -12

p7 = s9 * s10 = -2 * 14 = -84

Input Matrices: A & B

A = [1 3] B = [6 8]

[7 5] [4 2]

RESULT: C = A * B

C = [18 14]

[62 66]

c11 = p5 + p4 - p2 + p6 = 48 + (–10) – 8 + (–12) = 18

c12 = p1 + p2 = 6 + 8 = 14

c21 = p3 + p4 = 72 + (–10) = 62

c22 = p5 + p1 - p3 - p7 = 48 + 6 – 72 – (–84) = 66

$$A = \begin{bmatrix} 3 & 2 & 1 & 0 \\ 5 & 6 & 2 & 7 \\ 7 & 6 & 5 & 2 \\ 1 & 2 & 3 & 5 \end{bmatrix} \qquad A = \left[\begin{array}{cc|cc} 3 & 2 & 1 & 0 \\ 5 & 6 & 2 & 7 \\ \hline 7 & 6 & 5 & 2 \\ 1 & 2 & 3 & 5 \end{array}\right]$$

$$B = \begin{bmatrix} 5 & 6 & 7 & 1 \\ 1 & 3 & 2 & 0 \\ 5 & 6 & 7 & 1 \\ 2 & 3 & 9 & 0 \end{bmatrix} \qquad B = \left[\begin{array}{cc|cc} 5 & 6 & 7 & 1 \\ 1 & 3 & 2 & 0 \\ \hline 5 & 6 & 7 & 1 \\ 2 & 3 & 9 & 0 \end{array}\right]$$

$$A1 = \begin{bmatrix} 3 & 2 \\ 5 & 6 \end{bmatrix} \quad A2 = \begin{bmatrix} 1 & 0 \\ 2 & 7 \end{bmatrix} \qquad B1 = \begin{bmatrix} 5 & 6 \\ 1 & 3 \end{bmatrix} \quad B2 = \begin{bmatrix} 7 & 1 \\ 2 & 0 \end{bmatrix}$$

$$A3 = \begin{bmatrix} 7 & 6 \\ 1 & 2 \end{bmatrix} \quad A4 = \begin{bmatrix} 5 & 2 \\ 3 & 5 \end{bmatrix} \qquad B3 = \begin{bmatrix} 5 & 6 \\ 2 & 3 \end{bmatrix} \quad B4 = \begin{bmatrix} 7 & 1 \\ 9 & 0 \end{bmatrix}$$

$$A1 = \begin{bmatrix} 3 & 2 \\ 5 & 6 \end{bmatrix} \quad B1 = \begin{bmatrix} 5 & 6 \\ 1 & 3 \end{bmatrix} \quad C1 = \begin{bmatrix} 17 & 24 \\ 31 & 48 \end{bmatrix}$$

- P1= a*(f – h) = 3*(6-3) = 9
- P2 = h*(a + b) = 3*(3+2) = 15
- P3 = e*(c + d) = 5*(5+6) = 55
- P4 = d*(g – e) = 6*(1-5) = -24
- P5 = (a + d)*(e + h)= (3+6)*(5+3) = 72
- P6 = (b – d)*(g + h)= (2-6)*(1+3) = -16
- P7 = (a – c)*(e + f)= (3-5)+(5+6) = -22

$$C = \begin{bmatrix} P6 + P5 + P4 - P2 & P1 + P2 \\ P3 + P4 & P1 + P5 - P7 - P3 \end{bmatrix}$$

$$A2 = \begin{bmatrix} 1 & 0 \\ 2 & 7 \end{bmatrix} \quad B2 = \begin{bmatrix} 7 & 1 \\ 2 & 0 \end{bmatrix} \quad C2 = \begin{bmatrix} 7 & 1 \\ 28 & 2 \end{bmatrix}$$

- P1= a * (f – h) = 1
- P2 = h * (a + b) = 0
- P3 = e * (c + d) = 63
- P4 = d * (g – e) = -35
- P5 = (a + d) * (e + h) = 56
- P6 = (b – d) * (g + h) = -14
- P7 = (a – c) * (e + f) = -8

$$C = \begin{bmatrix} P6 + P5 + P4 - P2 & P1 + P2 \\ P3 + P4 & P1 + P5 - P7 - P3 \end{bmatrix}$$

$$A3 = \begin{bmatrix} 7 & 6 \\ 1 & 2 \end{bmatrix} \quad B3 = \begin{bmatrix} 5 & 6 \\ 2 & 3 \end{bmatrix} \quad C3 = \begin{bmatrix} 47 & 60 \\ 9 & 12 \end{bmatrix}$$

- P1= a * (f – h) = 21
- P2 = h * (a + b) = 39
- P3 = e * (c + d) = 15
- P4 = d * (g – e) = -6
- P5 = (a + d) * (e + h) = 72
- P6 = (b – d) * (g + h) = 20
- P7 = (a – c) * (e + f) = 66

$$C = \begin{bmatrix} P6 + P5 + P4 - P2 & P1 + P2 \\ P3 + P4 & P1 + P5 - P7 - P3 \end{bmatrix}$$

$$A4 = \begin{bmatrix} 5 & 2 \\ 3 & 5 \end{bmatrix} \quad B4 = \begin{bmatrix} 7 & 1 \\ 9 & 0 \end{bmatrix} \quad C4 = \begin{bmatrix} 59 & 5 \\ 66 & 3 \end{bmatrix}$$

- P1= a * (f – h) = 5
- P2 = h * (a + b) = 0
- P3 = e * (c + d) = 56
- P4 = d * (g – e) = 10
- P5 = (a + d) * (e + h) = 70
- P6 = (b – d) * (g + h) = -27
- P7 = (a – c) * (e + f) = 16

$$C = \begin{bmatrix} P6 + P5 + P4 - P2 & P1 + P2 \\ P3 + P4 & P1 + P5 - P7 - P3 \end{bmatrix}$$

$$A = \left[\begin{array}{cc|cc} 3 & 2 & 1 & 0 \\ 5 & 6 & 2 & 7 \\ \hline 7 & 6 & 5 & 2 \\ 1 & 2 & 3 & 5 \end{array}\right] \quad C1 = \begin{bmatrix} 17 & 24 \\ 31 & 48 \end{bmatrix} \quad C2 = \begin{bmatrix} 7 & 1 \\ 28 & 2 \end{bmatrix}$$

$$B = \left[\begin{array}{cc|cc} 5 & 6 & 7 & 1 \\ 1 & 3 & 2 & 0 \\ \hline 5 & 6 & 7 & 1 \\ 2 & 3 & 9 & 0 \end{array}\right] \quad C3 = \begin{bmatrix} 47 & 60 \\ 9 & 12 \end{bmatrix} \quad C4 = \begin{bmatrix} 59 & 5 \\ 66 & 3 \end{bmatrix}$$

Thank You