CSEN3001:
DESIGN AND ANALYSIS OF ALGORITHMS

UNIT-I: INTRODUCTION TO ALGORITHMS
Strassen’s Matrix Multiplication

Divide and conquer – general idea

- **Divide a problem into subprograms of the same kind**
- **Solve subprograms using the same approach**
- **Combine partial solution (if necessary)**

Matrix multiplication

Multiply two matrices A and B, each of size [n*n]

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}_{n\times n} \cdot \begin{bmatrix} & & \\ & B & \\ & & \end{bmatrix}_{n\times n} = \begin{bmatrix} & & \\ & C & \\ & & \end{bmatrix}_{n\times n}$$

Matrix multiplication – General Method

Mat C = Mat A * Mat B

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix} \qquad \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix}$$

Where,
c11=a11∗b11+a12∗b21+a13∗b31+a14∗b41

c12=a11∗b12+a12∗b22+a13∗b32+a14∗b42

c21=a21∗b11+a22∗b21+a23∗b31+a24∗b41

c22=a21∗b12+a22∗b22+a23∗b32+a24∗b42
So no….

$$C_{ij} = \sum_{k=1}^{n} A_{ik} \times B_{kj}$$

$$\therefore T(n) = O(n^3)$$

Matrix multiplication – Divide and Conquer Method

Transform the problem of multiplying *A* and *B*, each of size [n×n] into subproblems, each of size $\left[\frac{n}{2}\times\frac{n}{2}\right]$

$$\left[\begin{array}{ccc} C_{11} & & C_{12} \\ & C & \\ C_{21} & & C_{22} \end{array}\right] = \left[\begin{array}{ccc} A_{11} & & A_{12} \\ & A & \\ A_{21} & & A_{22} \end{array}\right] \cdot \left[\begin{array}{ccc} B_{11} & & B_{12} \\ & B & \\ B_{21} & & B_{22} \end{array}\right]$$

$$\boxed{\begin{aligned} \therefore T(n) &= 8\cdot T(\frac{n}{2}) + an^2 \\ &= O(n^3) \end{aligned}}$$

$$C_{11} = \underline{A_{11}\cdot B_{11}} + \underline{A_{12}\cdot B_{21}}$$

$$C_{12} = \underline{A_{11}\cdot B_{12}} + \underline{A_{12}\cdot B_{22}}$$

$$C_{21} = \underline{A_{21}\cdot B_{11}} + \underline{A_{22}\cdot B_{21}}$$

$$C_{22} = \underline{A_{21}\cdot B_{12}} + \underline{A_{22}\cdot B_{22}}$$

where an^2 is for addition. So, it is no improvement compared with the traditional way

Matrix multiplication – Divide and Conquer Method

$$\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}\cdot\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$$\begin{bmatrix}1&1&1&0\\1&1&1&0\\1&1&1&0\\0&0&0&0\end{bmatrix}\cdot\begin{bmatrix}1&1&1&0\\1&1&1&0\\1&1&1&0\\0&0&0&0\end{bmatrix}=\begin{bmatrix}3&3&3&0\\3&3&3&0\\3&3&3&0\\0&0&0&0\end{bmatrix}$$

$$C_{11}=\begin{bmatrix}1&1\\1&1\end{bmatrix}\cdot\begin{bmatrix}1&1\\1&1\end{bmatrix}+\begin{bmatrix}1&0\\1&0\end{bmatrix}\cdot\begin{bmatrix}1&1\\0&0\end{bmatrix}=\begin{bmatrix}3&3\\3&3\end{bmatrix}$$

$$C_{12}=\begin{bmatrix}1&1\\1&1\end{bmatrix}\cdot\begin{bmatrix}1&0\\1&0\end{bmatrix}+\begin{bmatrix}1&0\\1&0\end{bmatrix}\cdot\begin{bmatrix}1&0\\0&0\end{bmatrix}=\begin{bmatrix}3&0\\3&0\end{bmatrix}$$

$$C_{21}=\begin{bmatrix}1&1\\0&0\end{bmatrix}\cdot\begin{bmatrix}1&1\\1&1\end{bmatrix}+\begin{bmatrix}1&0\\0&0\end{bmatrix}\begin{bmatrix}1&1\\0&0\end{bmatrix}=\begin{bmatrix}3&3\\0&0\end{bmatrix}$$

$$C_{22}=\begin{bmatrix}1&1\\0&0\end{bmatrix}\cdot\begin{bmatrix}1&0\\1&0\end{bmatrix}+\begin{bmatrix}1&0\\0&0\end{bmatrix}\cdot\begin{bmatrix}1&0\\0&0\end{bmatrix}=\begin{bmatrix}3&0\\0&0\end{bmatrix}$$

Matrix multiplication – Strassen's Matrix Multiplication

- Discover a way to compute the C_{ij}'s using 7 multiplications and 18 additions or subtractions

$$P = (A_{11} + A_{22})(B_{11} + B_{22})$$

$$Q = (A_{21} + A_{22})B_{11}$$

$$R = A_{11}(B_{12} - B_{22})$$

$$S = A_{22}(B_{21} - B_{11})$$

$$T = (A_{11} + A_{12})B_{22}$$

$$U = (A_{21} - A_{11})(B_{11} + B_{12})$$

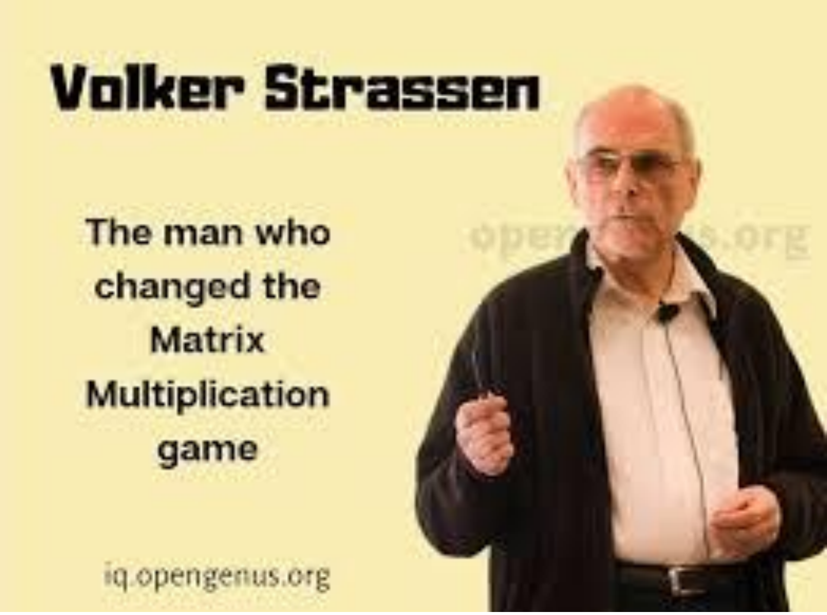
$$V = (A_{12} - A_{22})(B_{21} + B_{22})$$

$$C_{11} = P + S - T + V$$

$$C_{12} = R + T$$

$$C_{21} = Q + S$$

$$C_{22} = P + R - Q + U$$



Algorithm: Matrix Multiplication Strassen’s Method

```
Algorithm Strassen (n, A, B, C)
// n is size, A,B the input matrices, C output matrix
begin
        if n == 2
        {
                $C_{11} = a_{11} \cdot b_{11} + a_{12} \cdot b_{21}$
                $C_{12} = a_{11} \cdot b_{12} + a_{12} \cdot b_{22}$
                $C_{21} = a_{21} \cdot b_{11} + a_{22} \cdot b_{21}$
                $C_{22} = a_{21} \cdot b_{12} + a_{22} \cdot b_{22}$
        }
        else
        {
                Partition Matrix A into 4 submatrices
                ($A_{11}$, $A_{12}$, $A_{12}$, $A_{22}$)
                Partition Matrix B into 4 submatrices
                ($B_{11}$, $B_{12}$, $B_{21}$, $B_{22}$)
                call Strassen(n/2,$A_{11}$+$A_{22}$,$B_{11}$+$B_{22\,,}$P)
                call Strassen(n/2,$A_{21}$+$A_{22}$,$B_{11}$,Q)
                call Strassen(n/2,$A_{11}$,$B_{12}$-$B_{22}$,R)
                call Strassen(n/2,$A_{22}$,$B_{21}$- $B_{11}$,S)
                call Strassen(n/2,$A_{11}$+$A_{12}$,$B_{22}$,T)
                call Strassen(n/2,$A_{21}$+$A_{22}$,$B_{11}$,Q)
                call Strassen(n/2,$A_{21}$-$A_{11}$,$B_{11}$+ $B_{12}$,U)
                call Strassen(n/2,$A_{11}$-$A_{22}$,$B_{21}$+ $B_{22}$,V)

        }
end;
```

Algorithm: Matrix Multiplication Strassen's Method

$$\boxed{\begin{aligned} C_{11} &= P + S - T + V; \\ C_{12} &= R + T; \\ C_{21} &= Q + S; \\ C_{22} &= P + R - Q + U; \end{aligned}}$$

Matrix Multiplication Strassen's Method: Time Complexity

$$T(n) = \begin{cases} 7 \cdot T(\frac{n}{2}) + an^2 & n > 2 \\ b & n \le 2 \end{cases}$$

$$\begin{aligned} T(n) &= an^2[1 + 7/4 + (7/4)^2 + \cdots + (7/4)^{k-1}] + 7^k T(1) \\ &\le cn^2(7/4)^{\log_2 n} + 7^{\log_2 n},\ c \text{ a constant} \\ &= cn^{\log_2 4 + \log_2 7 - \log_2 4} + n^{\log_2 7} \\ &= O(n^{\log_2 7}) \approx O(n^{2.81}) \end{aligned}$$

Matrix Multiplication Strassen's Method: Time Complexity

$$T(n) = \begin{cases} 7 \cdot T(\frac{n}{2}) + an^2 & n > 2 \\ b & n \le 2 \end{cases}$$

$$\begin{aligned} T(n) &= 7 \cdot T(\frac{n}{2}) + an^2 \\ &= 7^2 \cdot T(\frac{n}{2^2}) + (\frac{7}{4})an^2 + an^2 \\ &= 7^3 \cdot T(\frac{n}{2^3}) + (\frac{7}{4})^2 \cdot an^2 + (\frac{7}{4}) \cdot an^2 + an^2 \\ &\cdots \end{aligned}$$

Matrix Multiplication Strassen’s Method: Time Complexity

Assume $n = 2^k$ for some integer k

$$= 7^{k-1} \cdot T(\frac{n}{2^{k-1}}) + an^2 \cdot \left[(\frac{7}{4})^{k-2} + \cdots + 1 \right]$$

$$= 7^{k-1} \cdot b + an^2 \left[\frac{(\frac{7}{4})^{k-1} - 1}{\frac{7}{4} - 1} \right]$$

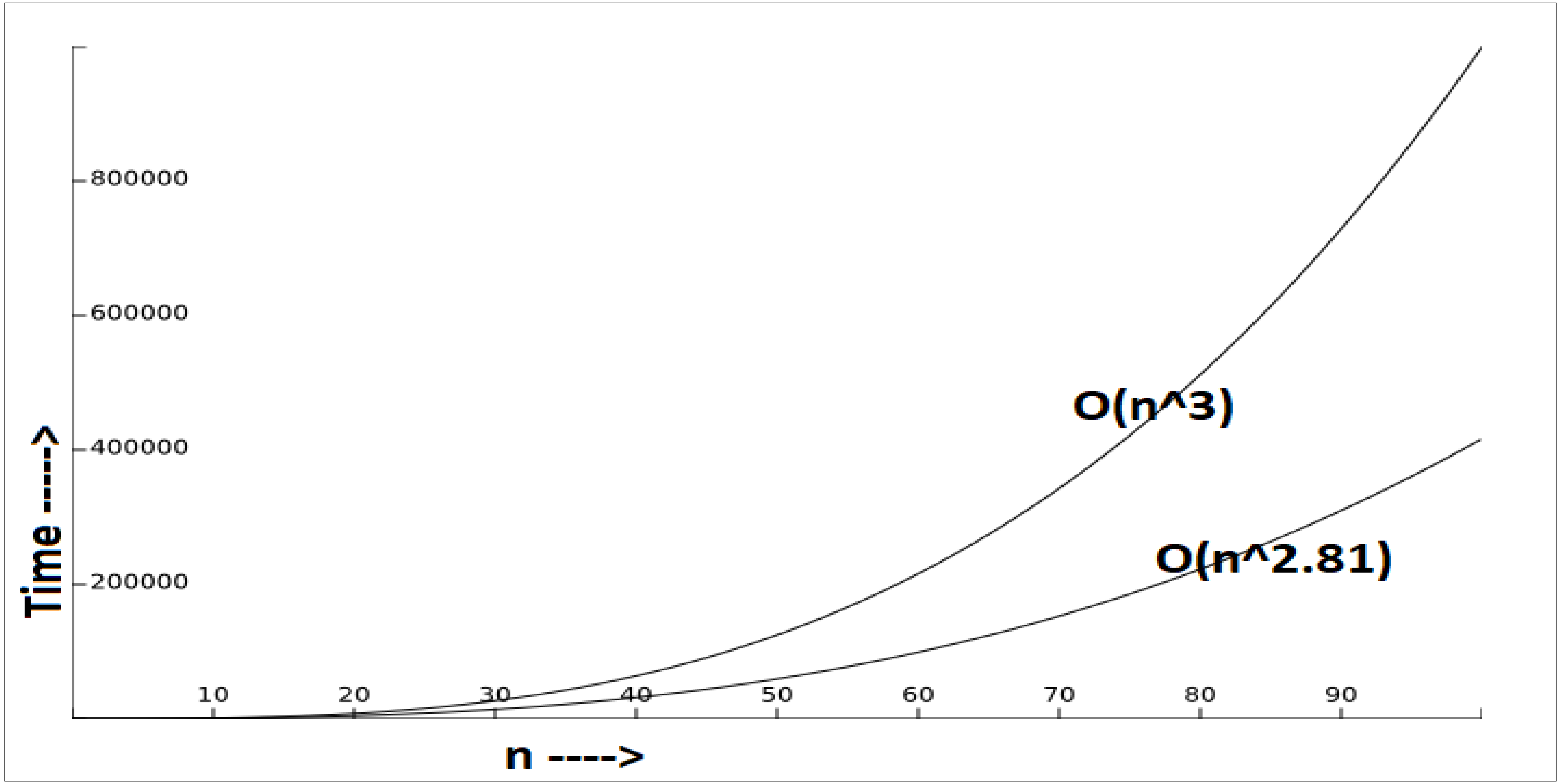
$$\leq b \cdot 7^k + c \cdot n^2 \cdot (\frac{7}{4})^k$$

$$= b \cdot 7^{Lgn} + cn^2 \cdot (\frac{7}{4})^{Lgn} = b \cdot 7^{Lgn} + cn^2 (n)^{Lg\frac{7}{4}}$$

$$= b \cdot n^{Lg7} + cn^{Lg7} = (b + c) \cdot n^{Lg7}$$

$$= O(n^{Lg7}) = O(n^{2.81})$$

Comparison : Straight method vs. Strassen's Method



Improvements and other methods

1978	**V. Y. Pan.** Strassen’s algorithm is not optimal.	**$O(n^{2.796})$**
1979	**D. Bini et al.** $O(n^{2.7799})$ complexity for nxn approximate matrix multiplication.	**$O(n^{2.7799})$**
1981	**A. Schönhage.** Partial and total matrix multiplication.	**$O(n^{2.522})$**

First to break the 2.5 barrier:

1981	**Copper Smith and Winograd** On the asymptotic complexity of matrix multiplication.	**$O(n^{2.496})$**
1986	**Volker Strassen**	**$O(n^{2.479})$**
1989	**CopperSmith and Winograd** Matrix multiplication via arithmetic progressions.	**$O(n^{2.376})$**
2011	**Virginia Vassilevska Williams**: Breaking the Coppersmith-Winograd barrier	**$O(n^{2.3727})$**

THANK YOU