STRASSEN's MATRIX MULTIPLICATION

① Conventional matrix multiplication method

Let A and B be two $n \times n$ matrices

$C = A \times B$ is also an $n \times n$ matrix

$$\boxed{c(i,j) = \sum_{1 \le k \le n} A(i,k) \times B(k,j)}$$

→ To compute $c[i,j]$ using this formula, we need n multiplications.

matrix c has $n \times n = n^2$ elements.

→ The time for the resulting matrix multiplication algorithm is $O(n^3)$

② The divide-and-conquer strategy suggests another way to compute the product of two $n \times n$ matrices.

→ For simplicity, we assume that n is a power of 2 (i.e., $n = 2^k$). In case if n is not power of 2, Then enough rows and columns of zeros can be added to both A and B so that the resulting dimensions are a power of two.

$$\begin{bmatrix} 10 & 5 & 8 \\ 6 & 4 & 9 \\ 15 & 3 & 20 \end{bmatrix}_{3\times3} \longrightarrow \begin{bmatrix} 10 & 5 & 8 & 0 \\ 6 & 4 & 9 & 0 \\ 15 & 3 & 20 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}_{4\times4}$$

$2^2 \times 2^2$

→ Imagine that A and B are each partitioned into 4 square submatrices having dimensions $\frac{n}{2} \times \frac{n}{2}$

→ If AB is $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \times \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$ Then

$$\boxed{\begin{aligned} C_{11} &= A_{11} B_{11} + A_{12} B_{21} \\ C_{12} &= A_{11} B_{12} + A_{12} B_{22} \\ C_{21} &= A_{21} B_{11} + A_{22} B_{21} \\ C_{22} &= A_{21} B_{12} + A_{22} B_{22} \end{aligned}} \longrightarrow (1)$$

→ This algorithm will continue applying itself to smaller-sized submatrices until n becomes suitably small (2×2) so that the product can be computed directly.

→ To compute AB using (1) we need to perform <u>8 multiplications</u> of $n/2 \times n/2$ matrices and <u>4 additions</u> of $n/2 \times n/2$ matrices.

→ Two $\frac{n}{2} \times \frac{n}{2}$ matrices can be added in time cn^2 for some constant c.

→ The overall computing time T(n) of the resulting divide-and-conquer algorithm is given by the recurrence relation

$$\boxed{T(n) = \begin{cases} b & \text{if } n \le 2 \\ 8T(n/2) + cn^2 & \text{if } n > 2 \end{cases}}$$

Derivation of Time Complexity:-

$$T(n) = 8\,T(n/2) + cn^2$$

$$= 8\left[8T(n/4) + c \cdot \frac{n^2}{4}\right] + cn^2$$

$$= 8^2\, T(n/4) + 3cn^2$$

$$= 8^2\left[8T(n/8) + c \cdot \frac{n^2}{4^2}\right] + 3cn^2$$

$$= 8^3\, T\left(\frac{n}{2^3}\right) + 7cn^2$$

At k^{th} step, we can write

$$T(n) = 8^k\, T(n/2^k) + (2^k - 1)\,cn^2$$

Substitute $n = 2^k \Rightarrow k = \log_2 n$

$$T(n) = 8^k\, T(n/n) + (n-1)\,cn^2$$

$$= (2^3)^k \times b + cn^3 - cn^2$$

$$= (2^k)^3 \times b + cn^3 - cn^2$$

$$T(n) = n^3 \times b + cn^3 - cn^2$$

$$T(n) \propto n^3$$

$$\boxed{\therefore T(n) = O(n^3)}$$

Again, we got same $O(n^3)$. No improvement over the conventional matrix multiplication method has been made. We can attempt to reformulate the equations for C_{ij} so as to have fewer multiplications and possibly more additions

Volker Strassen's method for matrix multiplication

Volker Strassen has discovered a way to compute the C_{ij}'s of equation (1) by using only 7 multiplications and 18 additions or subtractions. His method involves first computing the seven $\frac{n}{2} \times \frac{n}{2}$ submatrices, P, Q, R, S, T, U and V.

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

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

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

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

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

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

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

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

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

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

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

The resulting recurrence relation for $T(n)$ is

$$\boxed{T(n) = \begin{cases} b & \text{if } n \leq 2 \\ 7\,T\left(\frac{n}{2}\right) + \underset{\uparrow}{an^2} & \text{if } n > 2 \end{cases}}$$

$$18 \times \frac{n}{2} \times \frac{n}{2}$$

Time Complexity derivation :-

$$T(n) = \underline{7T(n/2)} + an^2$$

$$= 7\left[7T(n/4) + a\frac{n^2}{2^2}\right] + an^2$$

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

$$= 7^2\left[7T\left(\frac{n}{8}\right) + a \cdot \frac{n^2}{4^2}\right] + an^2\left[1 + \frac{7}{4}\right]$$

$$= 7^3 T\left(\frac{n}{2^3}\right) + an^2\left[1 + \frac{7}{4} + \frac{7^2}{4^2}\right]$$

$$\vdots$$

Similarly at k^{th} step, we can write

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

$$\approx 7^k T(n/n) + an^2\left(\frac{7}{4}\right)^k$$

$$\approx 7^k T(1) + an^2 \frac{7^k}{4^k}$$

$$\approx 7^k \times b + an^2 \frac{7^k}{4^k} \qquad (\because n = 2^k \quad k = \log_2 n)$$

$$\approx 7^{\log_2 n} \times b + an^2 \times \frac{7^{\log_2 n}}{4^{\log_2 n}}$$

$$\approx 7^{\log_2 n} \times b + an^2 \times \frac{7^{\log_2 n}}{n^{\log_2 4}}$$

$$\approx 7^{\log_2 n} \times b + a\cancel{n^2} \times \frac{7^{\log_2 n}}{\cancel{n^2}}$$

$$\approx 7^{\log_2 n}(a+b) = c \cdot n^{\log_2 7} = c \times n^{2.81}$$

$\boxed{\therefore T(n) = O(n^{2.81})}$ Time Complexity reduced from $O(n^3)$ to $O(n^{2.81})$