# Chapter - 4

**Divide & Conquer.**
- Divide
- Conquer
- Combine.

**Recurrences:**

It is an equation or Inequality that describes a function in term of its value on smaller inputs.

→ may take many form.

eg $T(n) = T(2n/3) + T(n/3) + \Theta(n)$

$T(n) = T(n-1) + \Theta(1)$

Solving recurrences (obtaining $\Theta$ or $O$ bounds.)

| Substitution Method | Recursion-tree Method | Master Method |
|---|---|---|
| • Guess a bound<br>• MI to Prove guess is Correct | Converts recurrences into tree whose nodes represent the cost incurred at various levels of recursion.<br>(Tech - bounding summation) | $T(n) = aT(n/b) + f(n)$<br>Sub Problem |

Recurrences that are Inequality.

$T(n) \le 2T(n/2) + \Theta(n)$ → Recurrences gives only upperbound on $T(n)$ i.e. O-notation is rep for solution.

$T(n) \geq 2T(n/2) + \Theta(n) \rightarrow$ lower bound
$\Omega(n)$

## STRASSEN'S ALGORITHUM.

↳ It is a Recursive algorithum for multiplying $n \times n$ matrices.

If $A = (a_{ij})$ and $B = b_{ij}$ ($n \times n$ matrices)

then product $C = A \cdot B$.

$$C_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

SQUARE-MATRIX-MULTIPLY - Matrix multiply
$O(n^3)$

$$T(n) = \begin{cases} \Theta(1) & , \text{if } n = 1 \\ 7T(n/2) + \Theta(n^2) & , \text{if } n > 1 \end{cases}$$

$S_1 = B_{12} - B_{22}$
$S_2 = A_{11} + A_{12}$
$S_3 = A_{21} + A_{22}$
$S_4 = B_{21} - B_{11}$
$S_5 = B_{11} + A_{22}$
$S_6 = B_{11} + B_{22}$
$S_7 = A_{12} + A_{22}$
$S_9 = A_{11} + A_{21}$
$S_8 = B_{21} + B_{22}$
$S_{10} = B_{11} + B_{12}$

$P_1 = A_{11} \cdot S_1$
$P_2 = S_2 \cdot B_{22}$
$P_3 = S_3 \cdot B_{11}$
$P_4 = A_{22} S_4$
$P_5 = S_5 \cdot S_6$
$P_6 = S_7 \cdot S_8$
$P_7 = S_9 \cdot S_{10}$

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

## Substitution Method

$T(n) = 2T(\lfloor n/2 \rfloor) + n$ ~~proves~~ $T(n) = O(n \log n)$

To Prove, $T(n) \leq c n \lg n$ for constant $c > 0$

$T(\lfloor n/2 \rfloor) \leq \lfloor n/2 \rfloor \log(\lfloor n/2 \rfloor)$

$$T(n) \leq 2\left(c \lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)\right) + n$$

$$\leq c n \lg \lfloor n/2 \rfloor + n$$

$$\leq c n \lg n - c n \lg 2 + n$$

$$= c n \log n - c n + n$$

$$\leq c n \lg n.$$

## Master's Theorem

$k \geq 0 \quad a \geq 1 \quad b > 1$

$$T(n) = aT(n/b) + O(n^k \log^p n)$$

**Case 1 Big O** $a > b^k$

$$T(n) = O(n^{\log_b a})$$

**Case 2 Theta $\Theta$** $a = b^k$

$p > -1 \quad T(n) = \Theta(n^{\log_b a} \log^p n)$

$p = -1 \quad T(n) = \Theta(n^{\log_b a} \log \log n)$

$p < -1 \quad T(n) = \Theta(n^{\log_b a})$

**Case 3 Omega $\Omega$** $a < b^k$

$p \geq 0 \quad T(n) = n^k \log^p n$

$p < 0 \quad T(n) = n^k.$

## Decision Tree

$T(n) = 3T(n/4) + Cn^2$ $\longrightarrow$ SubProblem size for a node at depth $i = n/4^i$

$\longrightarrow n/4^i = 1;\ n = 4^i$ $\quad i = \log_4 n.$

$Cn^2$

$T(n/4)\quad T(n/4)\quad T(n/4)$

$\longrightarrow Cn^2$

$Cn^2$

$C(n/4)^2 \cdot C(n/4)^2\ C(n/4)^2$ $\longrightarrow 3C\left(\frac{n}{4}\right)^2$

$T(n/16)\ T(n/16)\ T(n/16$ $\longrightarrow \left(\frac{3}{16}\right)^2 C\left(n/4\right)^2$

$O\left(n^{\log_4 3}\right)$

$T(1)\ T(1)\ T(1)\ T(1) \cdots\cdots$

$T(n) = 3T(n/4) + Cn^2$

$T(n/4) = 3T(n/16) + C(n/4)^2$

$T(n/16) = 3T(n/32) + C(n/16)^2$

Each level has 3 times more nodes than above level so no of nodes in level $i$ is $3^i$.

SubProblem reduces by a factor of 4 as we go down for $i = 0, 1, 2 \ldots \log_4 n - 1$

Depth of Tree is $\log_4 n$

ie levels $\Longrightarrow \log_4 n + 1)$

Each level has cost $\rightarrow c(n/4^i)^2$.

Multiplying total cost over all depth of the node $i$

$$\text{ie } 3^i c(n/4^i)^2 = (3/16)^i cn^2$$

A depth $\log_4 n$ $\Longrightarrow$ has $3^{\log_4 n} = n^{\log_4 3}$ nodes

$$\Rightarrow O(n^{\log_4 3})$$