Module 1

Recursion Tree

Master's Theorem

Recursion Tree Method

- Making a good guess is sometimes difficult with the substitution method.
- **Recursion Tree Method** can be used to devise a good guess.
- **Recursion Trees** show successive expansions of recurrences using trees.
- **Recursion Trees** model the costs (time) of a recursive execution of an algorithm that is composed of two part:
  - cost of non-recursive part.
  - cost of recursive call on smaller input size.
- A Tree node represents the cost of a sub-problem (recursive function invocation).
- To determine the total cost of the **Recursion Tree**, evaluate:
  - Cost of individual node at depth "*i*"
  - Sum up the cost of all nodes at depth "*i*"
  - Sum up all per-level costs of the **Recursion Tree**.

Recursion Tree Method: Example 1

$$T(n) = \begin{cases} 1 & n = 1 \\ 2T\left(\frac{n}{2}\right) + n & n > 1 \end{cases}$$

Solve the following recurrence using the Recurrence Tree Method.

Recursion Tree Method: Example 1

$$T(n) = \begin{cases} 1 & n = 1 \\ 2T\left(\frac{n}{2}\right) + n & n > 1 \end{cases}$$

Solve the following recurrence using the Recurrence Tree Method.

Assumption: We assume that n is exact power of 2.

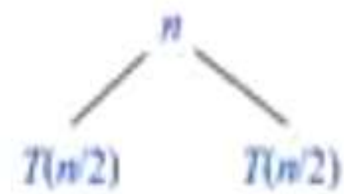
$$x^{\log_y n} \Rightarrow n^{\log_y x}$$

$$x^0 + x^1 + x^2 + \cdots + x^n = \frac{x^{n+1} - 1}{x - 1} \qquad for\ x \neq 1$$

$$x^0 + x^1 + x^2 + \cdots = \frac{1}{1 - x} \qquad for\ |x| < 1$$

Recursion Tree Method: Example 1

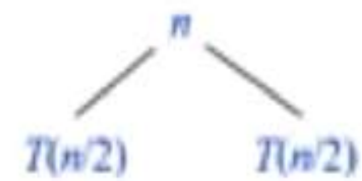
$T(n) = 2\,T\left(\frac{n}{2}\right) + n$



Recursion Tree Method: Example 1

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

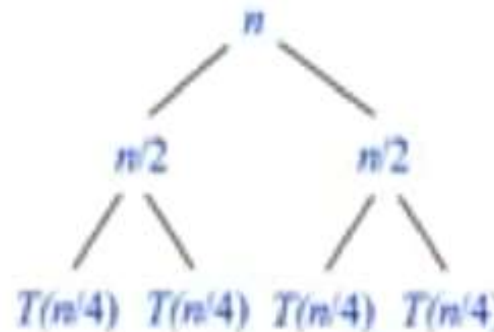
$$T\left(\frac{n}{2}\right) = 2.T\left(\frac{n}{2^2}\right) + \frac{n}{2}$$



Recursion Tree Method: Example 1

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

$$T\left(\frac{n}{2}\right) = 2.T\left(\frac{n}{2^2}\right) + \frac{n}{2}$$

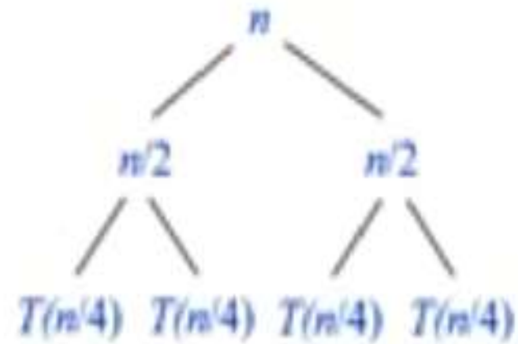


Recursion Tree Method: Example 1

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

$$T\left(\frac{n}{2}\right) = 2.T\left(\frac{n}{2^2}\right) + \frac{n}{2}$$

$$T\left(\frac{n}{2^2}\right) = 2.T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}$$



Recursion Tree Method: Example 1

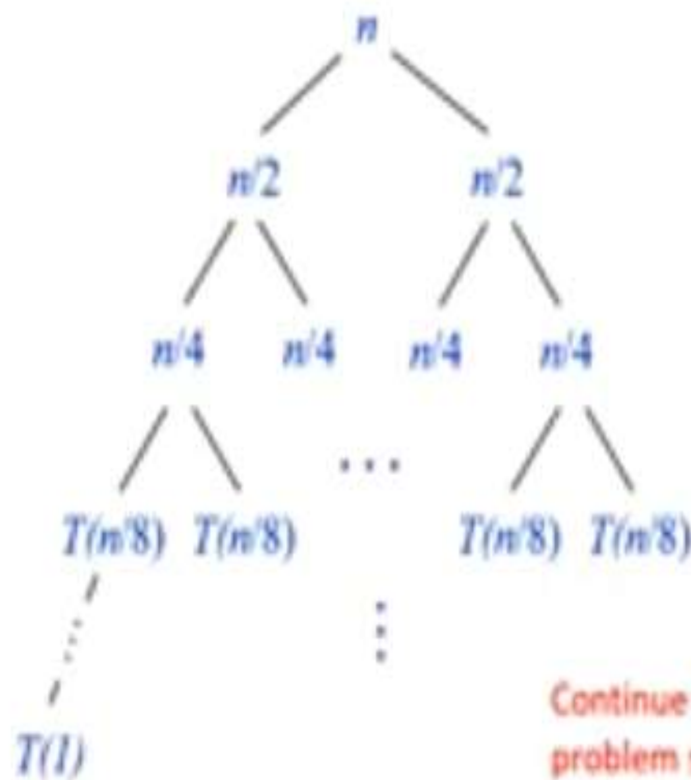
$$T(n) = 2\,T\left(\frac{n}{2}\right) + n$$

$$T\left(\frac{n}{2}\right) = 2.T\left(\frac{n}{2^2}\right) + \frac{n}{2}$$

$$T\left(\frac{n}{2^2}\right) = 2.T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}$$

$$T\left(\frac{n}{2^{k-1}}\right) = 2.T\left(\frac{n}{2^k}\right) + \frac{n}{2^{k-1}}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$



Continue expanding until the problem size reduces to 1.

Recursion Tree Method: Example 1

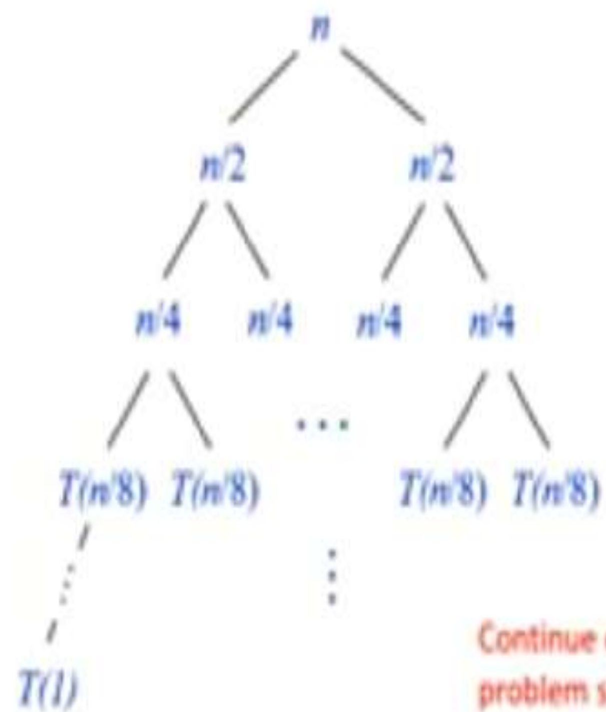
$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

$$T\left(\frac{n}{2}\right) = 2.T\left(\frac{n}{2^2}\right) + \frac{n}{2}$$

$$T\left(\frac{n}{2^2}\right) = 2.T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}$$

$$T\left(\frac{n}{2^{k-1}}\right) = 2.T\left(\frac{n}{2^k}\right) + \frac{n}{2^{k-1}}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$



Continue expanding until the problem size reduces to 1.

Total Cost = Cost of Leaf Nodes + Cost of Internal Nodes

Total Cost = (cost of leaf node x total leaf nodes) + (sum of costs at each level of internal nodes)

Total Cost = $L_n + I_n$

Recursion Tree Method: Example 1

$T(n) = 2T\left(\frac{n}{2}\right) + n$

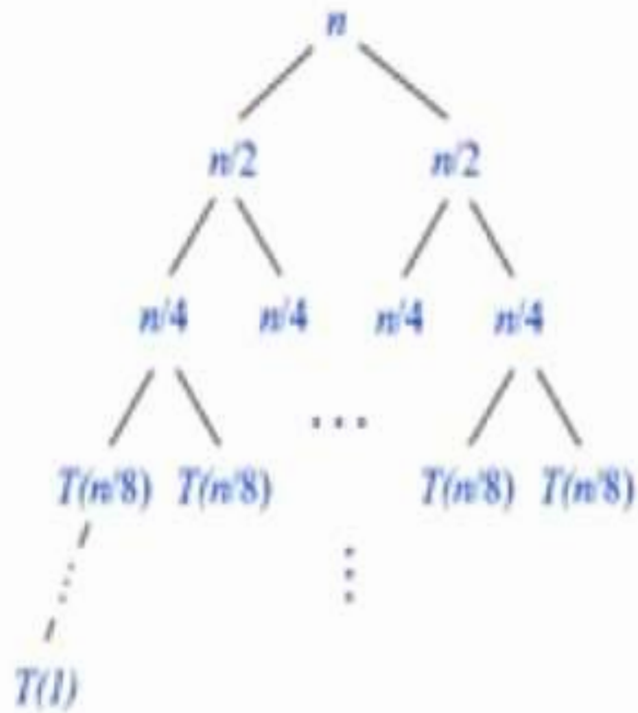
$T(\frac{n}{2}) = 2.T(\frac{n}{2^2}) + \frac{n}{2}$

$T(\frac{n}{2^2}) = 2.T(\frac{n}{2^3}) + \frac{n}{2^2}$

$T(\frac{n}{2^{k-1}}) = 2.T(\frac{n}{2^k}) + \frac{n}{2^{k-1}}$

$T(\frac{n}{2^k}) = T(1)$

$n = 2^k \Rightarrow k = \lg n$



Total Cost $= L_c + I_c$

Recursion Tree Method: Example 1

$T(n) = 2T\left(\frac{n}{2}\right) + n$

$T(\frac{n}{2}) = 2.T(\frac{n}{2^2}) + \frac{n}{2}$

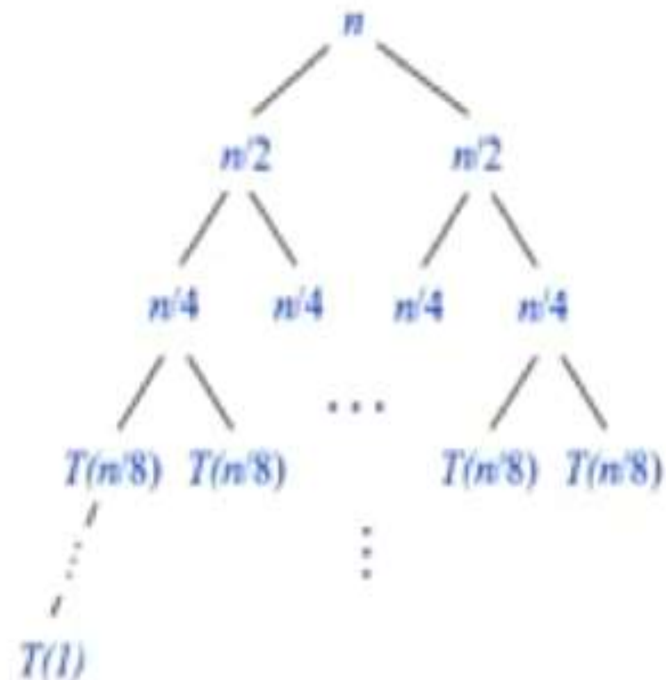
$T(\frac{n}{2^2}) = 2.T(\frac{n}{2^3}) + \frac{n}{2^2}$

$T(\frac{n}{2^{k-1}}) = 2.T(\frac{n}{2^k}) + \frac{n}{2^{k-1}}$

$T(\frac{n}{2^k}) = T(1)$

$n = 2^k \Rightarrow k = \lg n$

$L_c = 2^k \Rightarrow 2^{\lg n} \Rightarrow n^{\lg 2} \Rightarrow n$



Total Cost $= L_c + I_c$

Recursion Tree Method: Example 1

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

$$T\left(\frac{n}{2}\right) = 2.T\left(\frac{n}{2^2}\right) + \frac{n}{2}$$

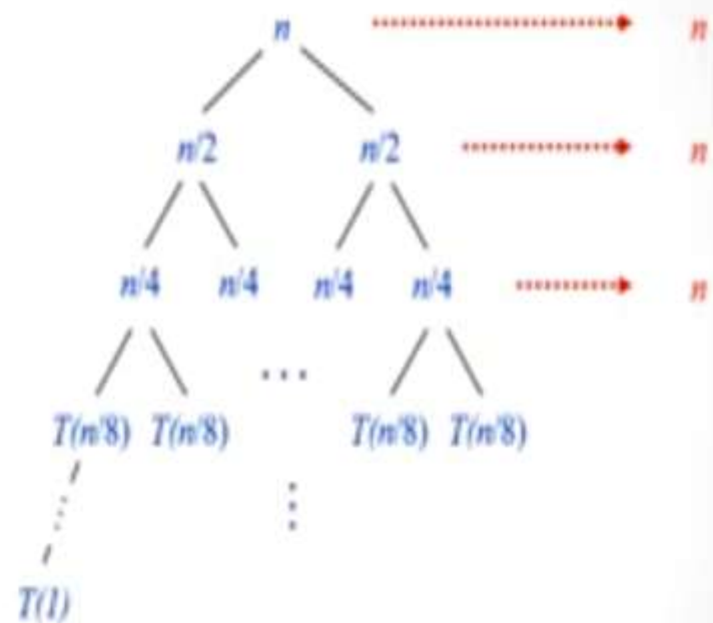
$$T\left(\frac{n}{2^2}\right) = 2.T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}$$

$$T\left(\frac{n}{2^{k-1}}\right) = 2.T\left(\frac{n}{2^k}\right) + \frac{n}{2^{k-1}}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$

$$n = 2^k \Longrightarrow k = \lg n$$

$$L_c = 2^k \Longrightarrow 2^{\lg n} \Longrightarrow n^{\lg 2} \Longrightarrow n$$



Total Cost $= L_c + I_c$

Recursion Tree Method: Example 1

$$T(n) = 2\,T\left(\frac{n}{2}\right) + n$$

$$T\left(\frac{n}{2}\right) = 2.T\left(\frac{n}{2^2}\right) + \frac{n}{2}$$

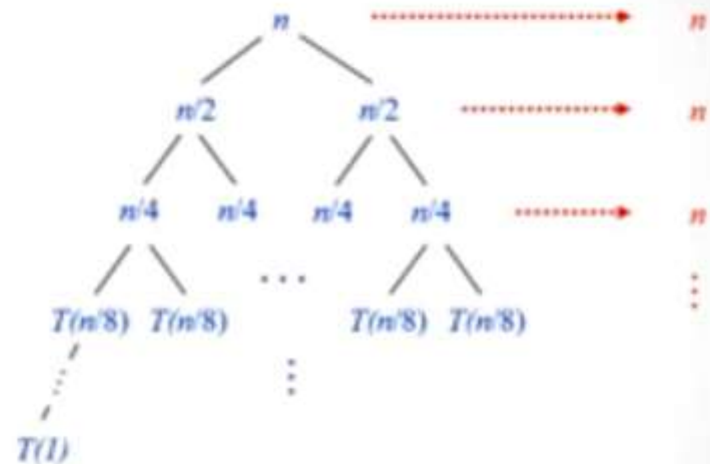
$$T\left(\frac{n}{2^2}\right) = 2.T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}$$

$$T\left(\frac{n}{2^{k-1}}\right) = 2.T\left(\frac{n}{2^k}\right) + \frac{n}{2^{k-1}}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$

$$n = 2^k \Longrightarrow k = \lg n$$

$$L_c = 2^k \Longrightarrow 2^{\lg n} \Longrightarrow n^{\lg 2} \Longrightarrow n$$



$$I_c = k \cdot n$$

$$I_c = n \lg n$$

Total Cost $= L_c + I_c \Longrightarrow n + n\lg n$

Recursion Tree Method: Example 1

$T(n) = 2T\left(\frac{n}{2}\right) + n$

$T(\frac{n}{2}) = 2.T(\frac{n}{2^2}) + \frac{n}{2}$

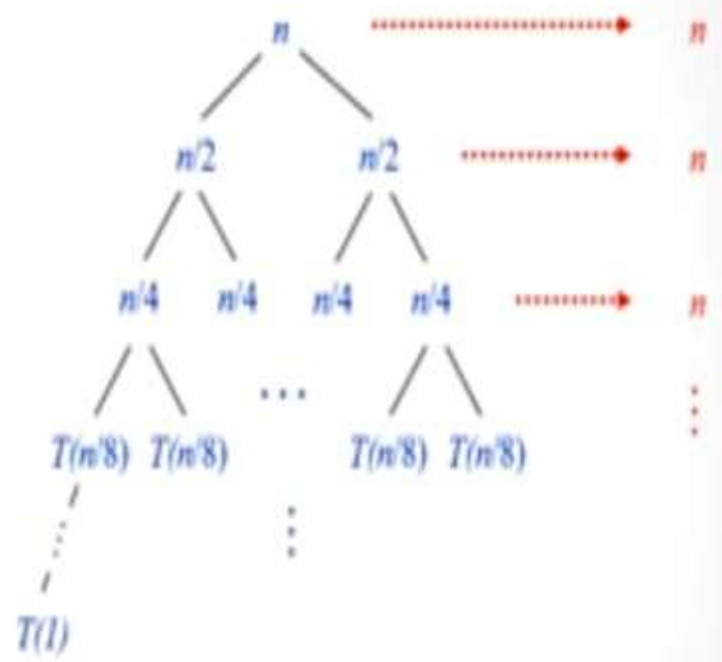
$T(\frac{n}{2^2}) = 2.T(\frac{n}{2^3}) + \frac{n}{2^2}$

$T(\frac{n}{2^{k-1}}) = 2.T(\frac{n}{2^k}) + \frac{n}{2^{k-1}}$

$T(\frac{n}{2^k}) = T(1)$

$n = 2^k \implies k = \lg n$

$L_c = 2^k \implies 2^{\lg n} \implies n^{\lg 2} \implies n$



$I_c = k \cdot n$

$I_c = n \lg n$

Total Cost $= L_c + I_c \implies n + n \lg n$

Hence: $T(n) \in O(n \lg n)$

Recursion Tree Method: Example 2

$$T(n) = \begin{cases} 1 & n = 1 \\ 2T\left(\frac{n}{2}\right) + n^2 & n > 1 \end{cases}$$

Solve the following recurrence using the Recurrence Tree Method.

Recursion Tree Method: Example 2

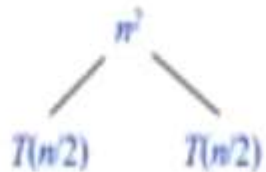
$$T(n) = \begin{cases} 1 & n = 1 \\ 2T\left(\frac{n}{2}\right) + n^2 & n > 1 \end{cases}$$

Solve the following recurrence using the Recurrence Tree Method.

Assumption: We assume that n is exact power of 2.

Recursion Tree Method: Example 2

$$T(n) = 2T\left(\frac{n}{2}\right) + n^2$$

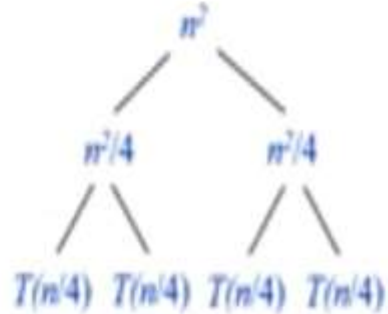


Recursion Tree Method: Example 2

$$T(n) = 2T\left(\frac{n}{2}\right) + n^2$$

$$T\left(\frac{n}{2}\right) = 2.T\left(\frac{n}{2^2}\right) + \frac{n^2}{2^2}$$

$$T\left(\frac{n}{2^2}\right) = 2.T\left(\frac{n}{2^3}\right) + \frac{n^2}{4^2}$$



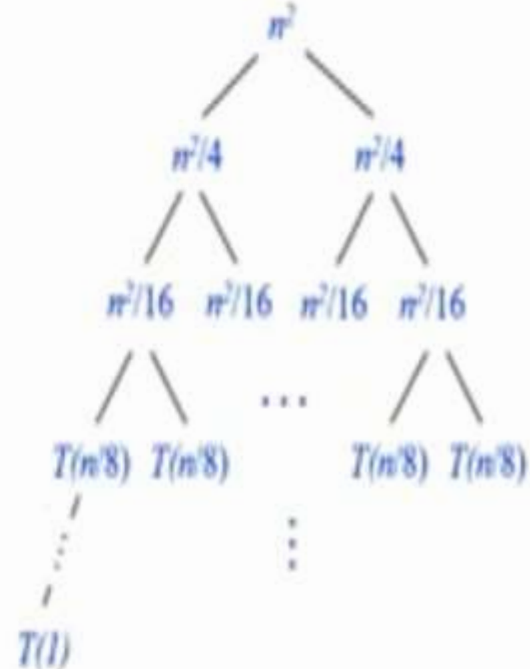
Recursion Tree Method: Example 2

$$T(n) = 2T\left(\frac{n}{2}\right) + n^2$$

$$T\left(\frac{n}{2}\right) = 2.T\left(\frac{n}{2^2}\right) + \frac{n^2}{2^2}$$

$$T\left(\frac{n}{2^2}\right) = 2.T\left(\frac{n}{2^3}\right) + \frac{n^2}{4^2}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$



Recursion Tree Method: Example 2

$T(n) = 2T\left(\frac{n}{2}\right) + n^2$

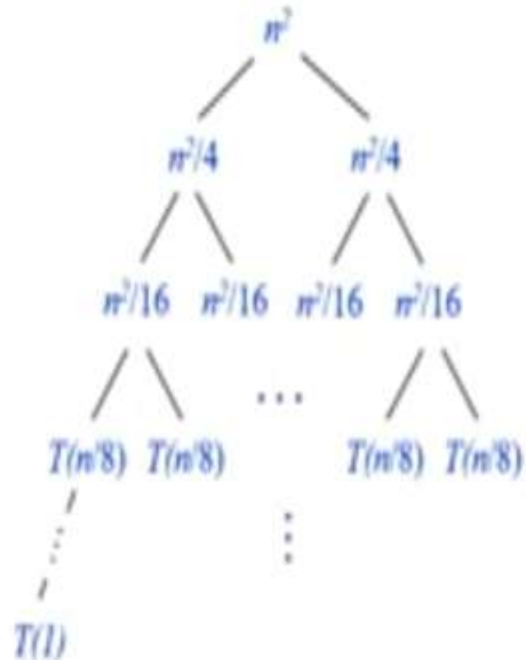
$T(\frac{n}{2}) = 2.T(\frac{n}{2^2}) + \frac{n^2}{2^2}$

$T(\frac{n}{2^2}) = 2.T(\frac{n}{2^3}) + \frac{n^2}{4^2}$

$T(\frac{n}{2^k}) = T(1)$

$n = 2^k \implies k = \lg n$

$L_c = 2^k \implies 2^{\lg n} \implies n^{\lg 2} \implies n$



Total Cost $= L_c + I_c$

Recursion Tree Method: Example 2

$$T(n) = 2T\left(\frac{n}{2}\right) + n^2$$

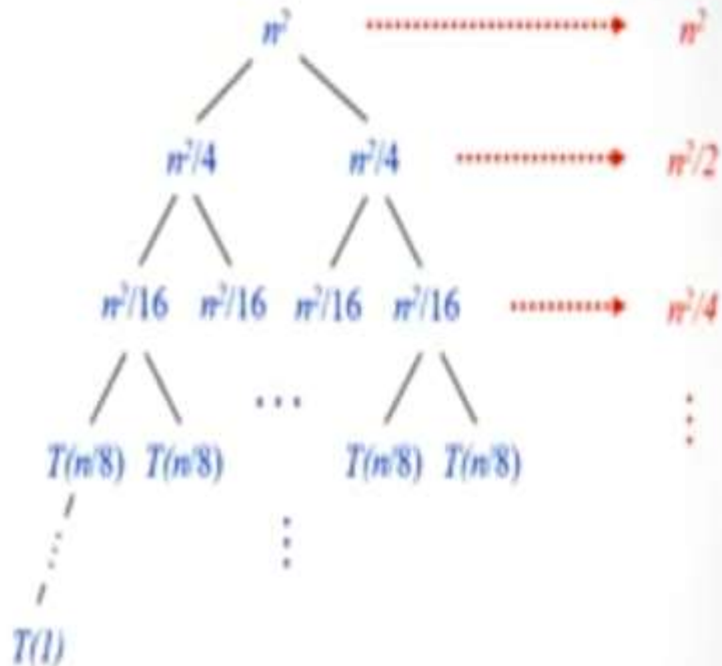
$$T\left(\frac{n}{2}\right) = 2.T\left(\frac{n}{2^2}\right) + \frac{n^2}{2^2}$$

$$T\left(\frac{n}{2^2}\right) = 2.T\left(\frac{n}{2^3}\right) + \frac{n^2}{4^2}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$

$$n = 2^k \Longrightarrow k = \lg n$$

$$L_c = 2^k \Longrightarrow 2^{\lg n} \Longrightarrow n^{\lg 2} \Longrightarrow n$$



Total Cost $= L_c + I_c$

Recursion Tree Method: Example 2

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

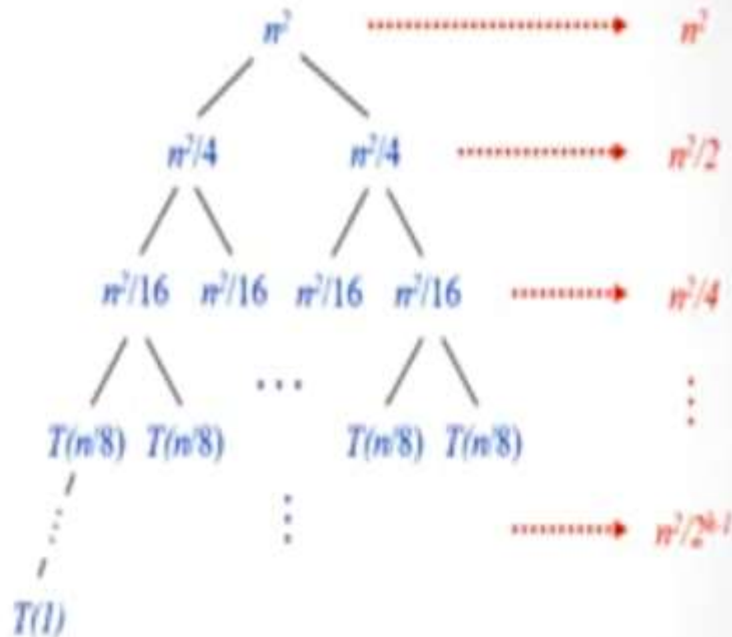
$$T\left(\frac{n}{2}\right) = 2.T\left(\frac{n}{2^2}\right) + \frac{n^2}{2^2}$$

$$T\left(\frac{n}{2^2}\right) = 2.T\left(\frac{n}{2^3}\right) + \frac{n^2}{4^2}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$

$$n = 2^k \Longrightarrow k = \lg n$$

$$L_c = 2^k \longrightarrow 2^{\lg n} \longrightarrow n^{\lg 2} \longrightarrow n$$



Total Cost $= L_c + I_c$

Recursion Tree Method: Example 2

$$T(n) = 2T\left(\frac{n}{2}\right) + n^2$$

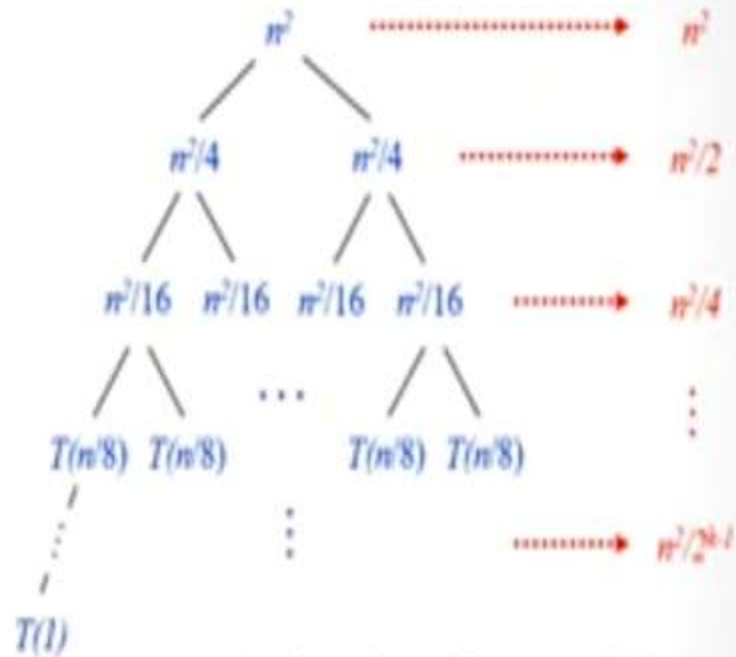
$$T\left(\frac{n}{2}\right) = 2.T\left(\frac{n}{2^2}\right) + \frac{n^2}{2^2}$$

$$T\left(\frac{n}{2^2}\right) = 2.T\left(\frac{n}{2^3}\right) + \frac{n^2}{4^2}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$

$$n = 2^k \Rightarrow k = \lg n$$

$$L_c = 2^k \Rightarrow 2^{\lg n} \Rightarrow n^{\lg 2} \Rightarrow n$$



$$I_c = n^2 \cdot \left[\left(\frac{1}{2}\right)^0 + \left(\frac{1}{2}\right)^1 + \left(\frac{1}{2}\right)^2 + \cdots + \left(\frac{1}{2}\right)^{k-1}\right]$$

Total Cost $= L_c + I_c$

Recursion Tree Method: Example 2

$$T(n) = 2T\left(\frac{n}{2}\right) + n^2$$

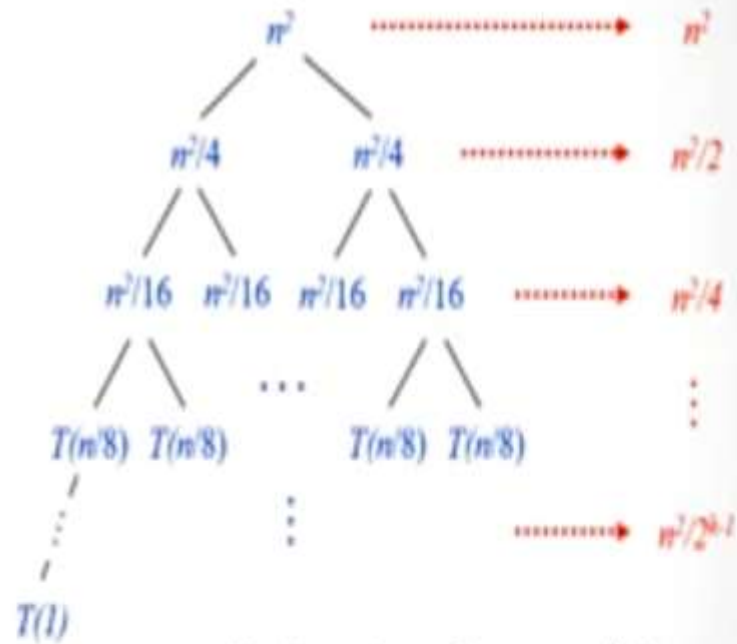
$$T\left(\frac{n}{2}\right) = 2.T\left(\frac{n}{2^2}\right) + \frac{n^2}{2^2}$$

$$T\left(\frac{n}{2^2}\right) = 2.T\left(\frac{n}{2^3}\right) + \frac{n^2}{4^2}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$

$$n = 2^k \Rightarrow k = \lg n$$

$$L_c = 2^k \Rightarrow 2^{\lg n} \Rightarrow n^{\lg 2} \Rightarrow n$$



$$I_c = n^2 \cdot \left[\left(\tfrac{1}{2}\right)^0 + \left(\tfrac{1}{2}\right)^1 + \left(\tfrac{1}{2}\right)^2 + \cdots + \left(\tfrac{1}{2}\right)^{k-1}\right]$$

$$I_c = n^2 \cdot \left[\frac{1}{1 - {}^1/_2}\right] \Rightarrow 2n^2$$

Total Cost $= L_c + I_c$

Recursion Tree Method: Example 2

$$T(n) = 2T\left(\frac{n}{2}\right) + n^2$$

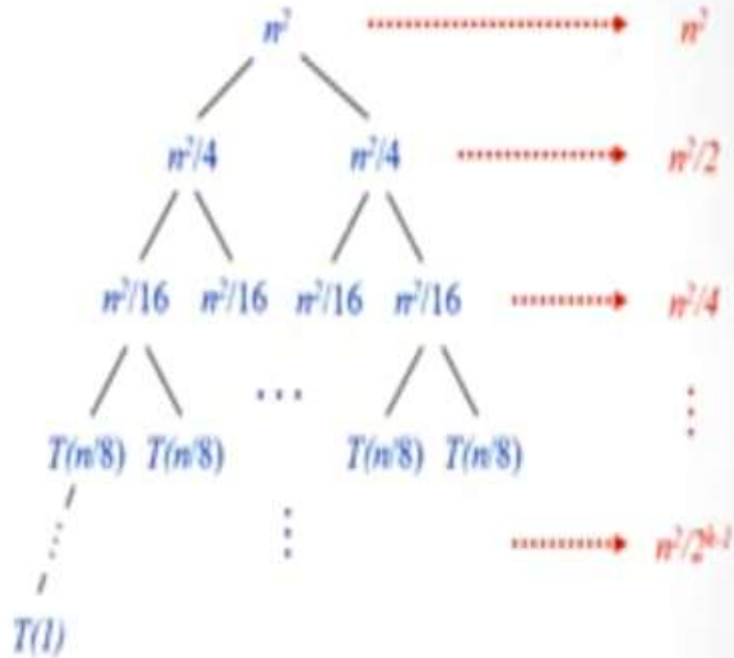
$$T\left(\frac{n}{2}\right) = 2.T\left(\frac{n}{2^2}\right) + \frac{n^2}{2^2}$$

$$T\left(\frac{n}{2^2}\right) = 2.T\left(\frac{n}{2^3}\right) + \frac{n^2}{4^2}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$

$$n = 2^k \Rightarrow k = \lg n$$

$$L_c = 2^k \Rightarrow 2^{\lg n} \Rightarrow n^{\lg 2} \Rightarrow n$$



$$I_c = n^2 \cdot \left[\left(\frac{1}{2}\right)^0 + \left(\frac{1}{2}\right)^1 + \left(\frac{1}{2}\right)^2 + \cdots + \left(\frac{1}{2}\right)^{k-1}\right]$$

$$I_c = n^2 \cdot \left[\frac{1}{1 - \frac{1}{2}}\right] \Rightarrow 2n^2$$

Total Cost $= L_c + I_c \Rightarrow n + 2n^2$

Hence: $T(n) \in O(n^2)$

Recursion Tree Method: Example 3

$$T(n) = \begin{cases} 1 & n = 1 \\ 3T\left(\frac{n}{4}\right) + n^2 & n > 1 \end{cases}$$

Solve the following recurrence using the Recurrence Tree Method.

Recursion Tree Method: Example 3

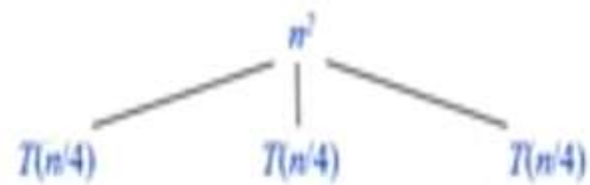
$$T(n) = \begin{cases} 1 & n = 1 \\ 3T\left(\frac{n}{4}\right) + n^2 & n > 1 \end{cases}$$

Solve the following recurrence using the Recurrence Tree Method.

Assumption: We assume that n is exact power of 4.

Recursion Tree Method: Example 3

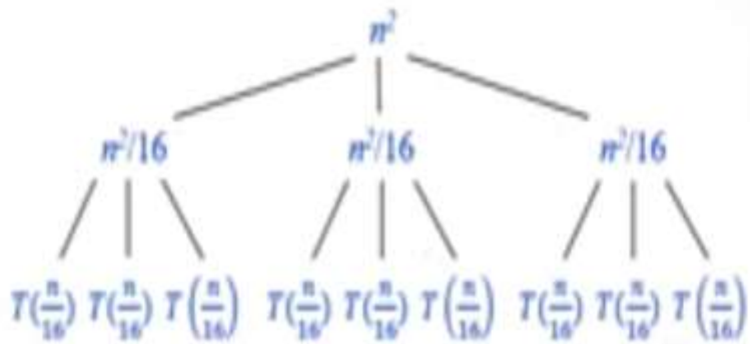
$$T(n) = 3T\left(\frac{n}{4}\right) + n^2$$



Recursion Tree Method: Example 3

$$T(n) = 3\,T\left(\frac{n}{4}\right) + n^2$$

$$T\left(\frac{n}{4}\right) = 3.T\left(\frac{n}{4^2}\right) + \frac{n^2}{4^2}$$

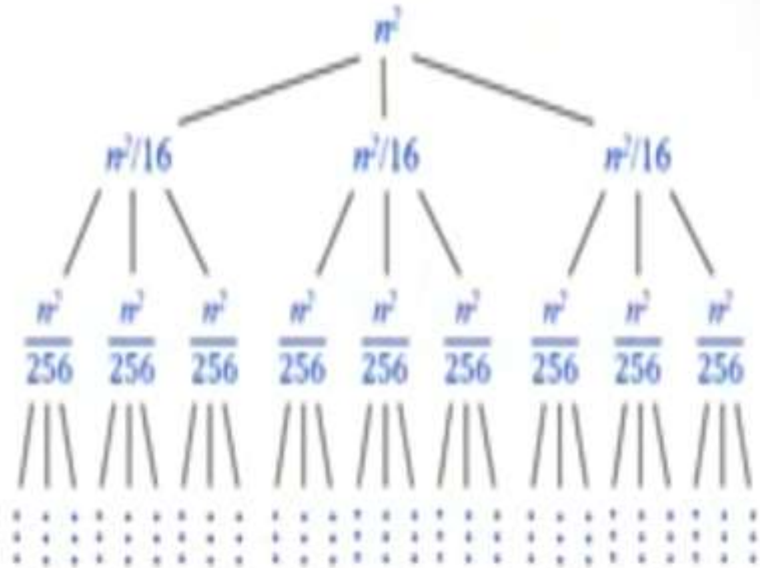


Recursion Tree Method: Example 3

$$T(n) = 3T\left(\frac{n}{4}\right) + n^2$$

$$T\left(\frac{n}{4}\right) = 3.T\left(\frac{n}{4^2}\right) + \frac{n^2}{4^2}$$

$$T\left(\frac{n}{4^2}\right) = 3.T\left(\frac{n}{4^3}\right) + \frac{n^2}{16^2}$$



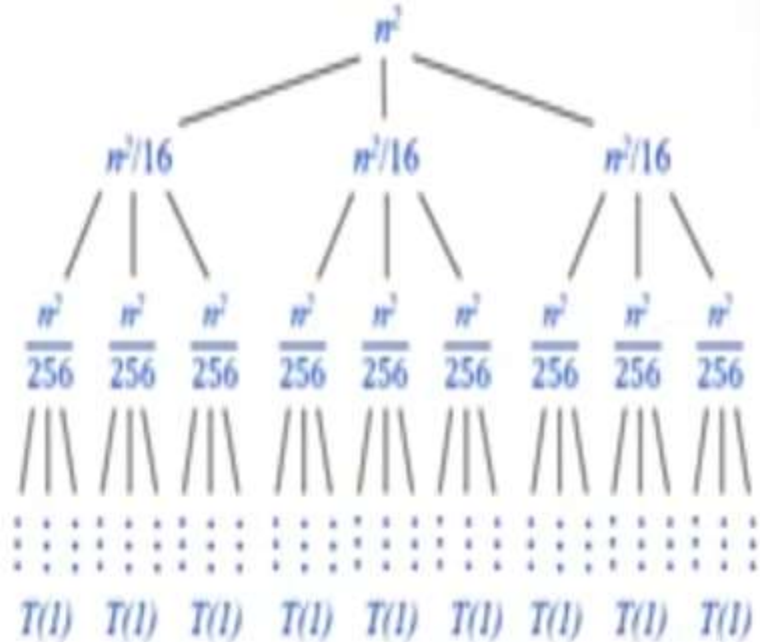
Recursion Tree Method: Example 3

$$T(n) = 3T\left(\frac{n}{4}\right) + n^2$$

$$T\left(\frac{n}{4}\right) = 3.T\left(\frac{n}{4^2}\right) + \frac{n^2}{4^2}$$

$$T\left(\frac{n}{4^2}\right) = 3.T\left(\frac{n}{4^3}\right) + \frac{n^2}{16^2}$$

$$T\left(\frac{n}{4^k}\right) = T(1)$$



Total Cost = $L_c + I_c$

Recursion Tree Method: Example 3

$T(n) = 3T\left(\frac{n}{4}\right) + n^2$

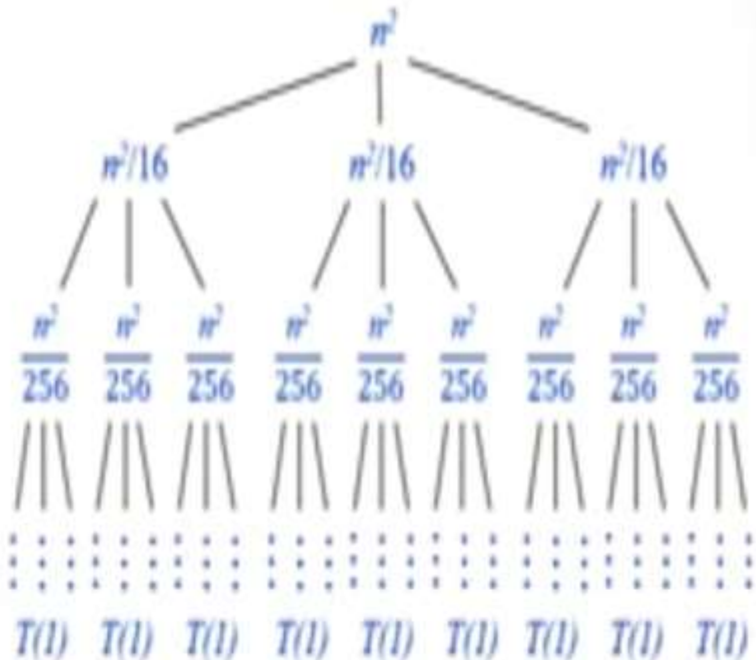
$T\left(\frac{n}{4}\right) = 3.T\left(\frac{n}{4^2}\right) + \frac{n^2}{4^2}$

$T\left(\frac{n}{4^2}\right) = 3.T\left(\frac{n}{4^3}\right) + \frac{n^2}{16^2}$

$T\left(\frac{n}{4^k}\right) = T(1)$

$n = 4^k$

$k = \log_4 n$



Total Cost $= L_n + I_n$

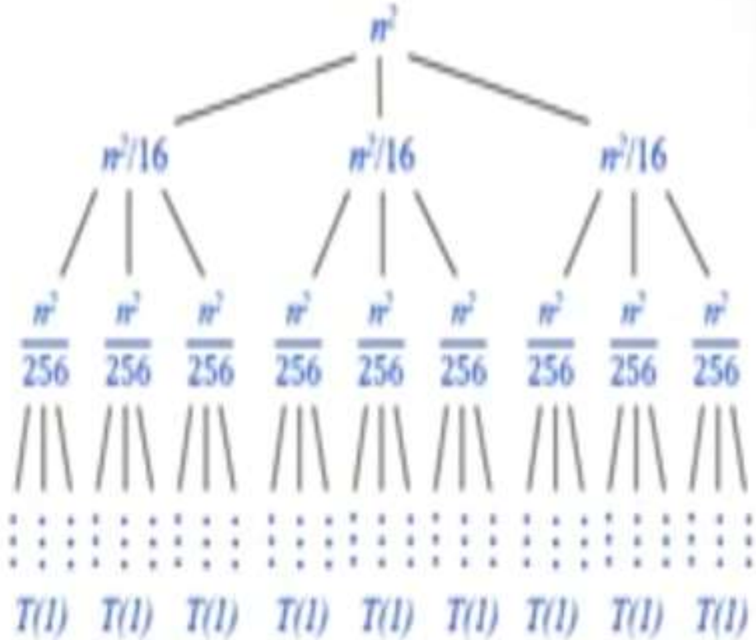
Recursion Tree Method: Example 3

$T(n) = 3\,T\left(\frac{n}{4}\right) + n^2$

$T(\frac{n}{4}) = 3.T(\frac{n}{4^2}) + \frac{n^2}{4^2}$

$T(\frac{n}{4^2}) = 3.T(\frac{n}{4^3}) + \frac{n^2}{16^2}$

$T(\frac{n}{4^k}) = T(1)$



$n = 4^k$

$k = \log_4 n$

$L_c = 3^k \implies 3^{\log_4 n}$

Total Cost $= L_c + I_c$

Recursion Tree Method: Example 3

$$T(n) = 3T\left(\frac{n}{4}\right) + n^2$$

$$T\left(\frac{n}{4}\right) = 3.T\left(\frac{n}{4^2}\right) + \frac{n^2}{4^2}$$

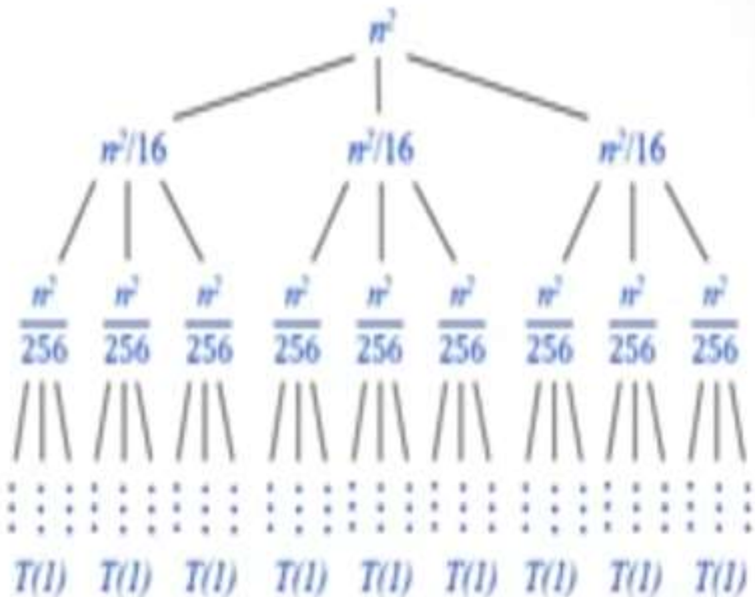
$$T\left(\frac{n}{4^2}\right) = 3.T\left(\frac{n}{4^3}\right) + \frac{n^2}{16^2}$$

$$T\left(\frac{n}{4^k}\right) = T(1)$$

$n = 4^k$

$k = \log_4 n$

$L_c = 3^k \implies 3^{\log_4 n} \implies n^{\log_4 3}$



Total Cost = $L_c + I_c$

Recursion Tree Method: Example 3

$T(n) = 3\,T\left(\frac{n}{4}\right) + n^2$

$T\left(\frac{n}{4}\right) = 3.T\left(\frac{n}{4^2}\right) + \frac{n^2}{4^2}$

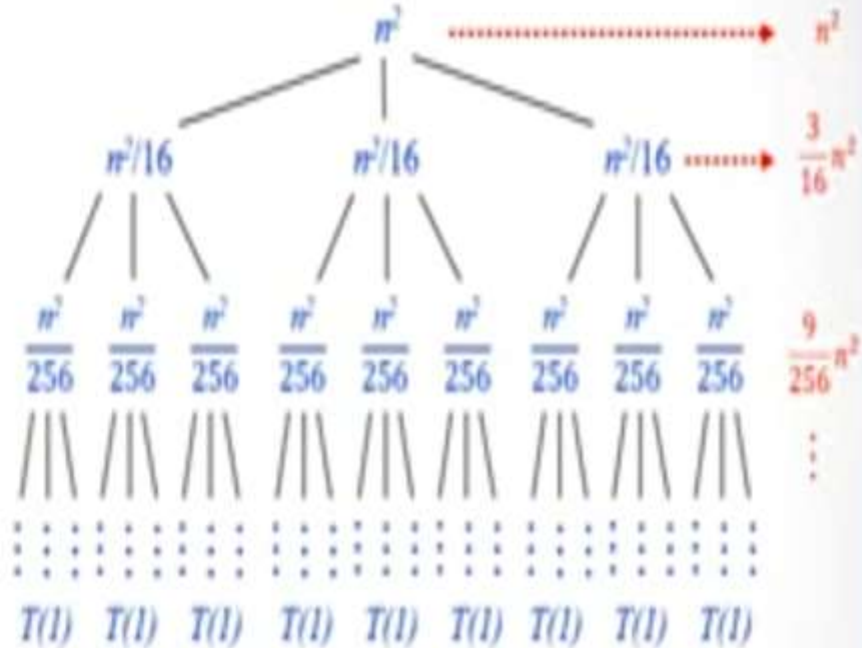
$T\left(\frac{n}{4^2}\right) = 3.T\left(\frac{n}{4^3}\right) + \frac{n^2}{16^2}$

$T\left(\frac{n}{4^k}\right) = T(1)$

$n = 4^k$

$k = \log_4 n$

$L_c = 3^k \Longrightarrow 3^{\log_4 n} \Longrightarrow n^{\log_4 3}$



Total Cost = $L_c + I_c$

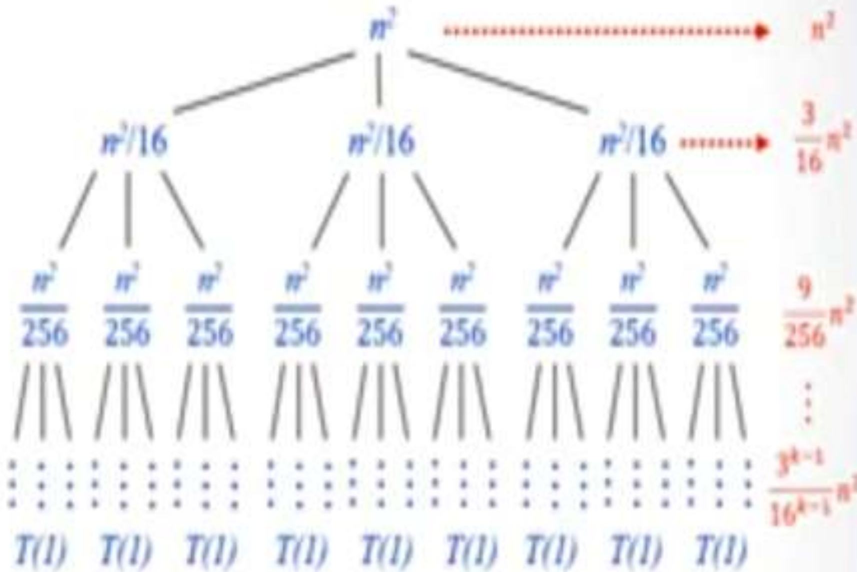
Recursion Tree Method: Example 3

$$T(n) = 3T\left(\frac{n}{4}\right) + n^2$$

$$T\left(\frac{n}{4}\right) = 3.T\left(\frac{n}{4^2}\right) + \frac{n^2}{4^2}$$

$$T\left(\frac{n}{4^2}\right) = 3.T\left(\frac{n}{4^3}\right) + \frac{n^2}{16^2}$$

$$T\left(\frac{n}{4^k}\right) = T(1)$$



$n = 4^k$

$k = \log_4 n$

$L_c = 3^k \implies 3^{\log_4 n} \implies n^{\log_4 3}$

$$I_c = n^2 \cdot \left[\left(\frac{3}{16}\right)^0 + \left(\frac{3}{16}\right)^1 + \left(\frac{3}{16}\right)^2 + \cdots + \left(\frac{3}{16}\right)^{k-1}\right]$$

Total Cost $= L_c + I_c$

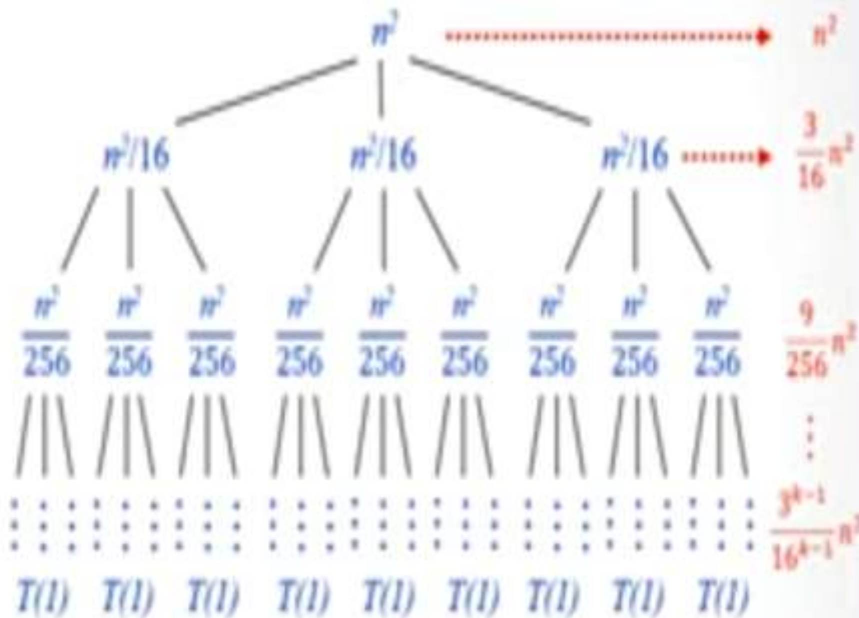
Recursion Tree Method: Example 3

$$T(n) = 3\,T\left(\frac{n}{4}\right) + n^2$$

$$T\left(\frac{n}{4}\right) = 3.T\left(\frac{n}{4^2}\right) + \frac{n^2}{4^2}$$

$$T\left(\frac{n}{4^2}\right) = 3.T\left(\frac{n}{4^3}\right) + \frac{n^2}{16^2}$$

$$T\left(\frac{n}{4^k}\right) = T(1)$$



$n = 4^k$

$k = \log_4 n$

$L_c = 3^k \Rightarrow 3^{\log_4 n} \Rightarrow n^{\log_4 3}$

$$I_c = n^2 \cdot \left[\left(\frac{3}{16}\right)^0 + \left(\frac{3}{16}\right)^1 + \left(\frac{3}{16}\right)^2 + \cdots + \left(\frac{3}{16}\right)^{k-1}\right]$$

$$I_c = n^2 \cdot \left[\frac{1}{1 - {}^3/_{16}}\right] \Rightarrow \frac{16}{13}n^2$$

Total Cost $= L_c + I_c$

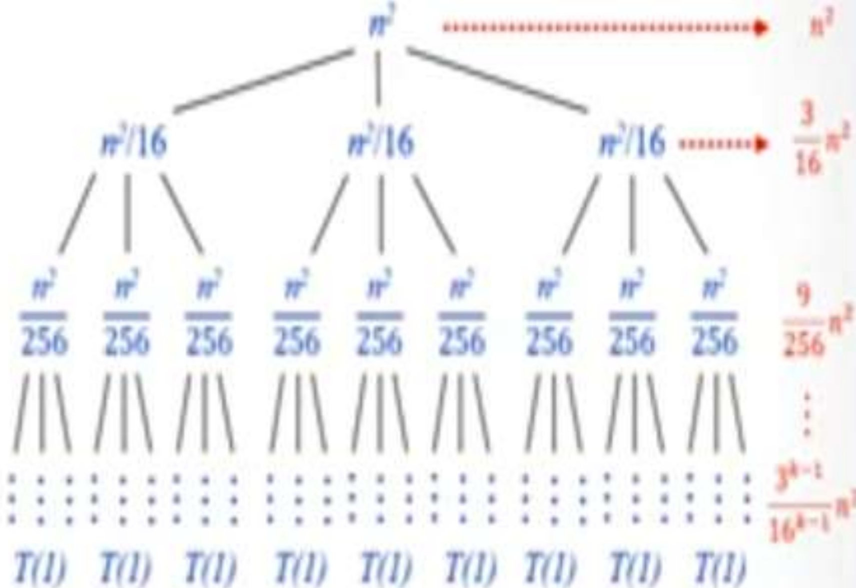
Recursion Tree Method: Example 3

$T(n) = 3T\left(\frac{n}{4}\right) + n^2$

$T\left(\frac{n}{4}\right) = 3.T\left(\frac{n}{4^2}\right) + \frac{n^2}{4^2}$

$T\left(\frac{n}{4^2}\right) = 3.T\left(\frac{n}{4^3}\right) + \frac{n^2}{16^2}$

$T\left(\frac{n}{4^k}\right) = T(1)$



$n = 4^k$

$k = \log_4 n$

$L_c = 3^k \implies 3^{\log_4 n} \implies n^{\log_4 3}$

$I_c = n^2 \cdot \left[\left(\frac{3}{16}\right)^0 + \left(\frac{3}{16}\right)^1 + \left(\frac{3}{16}\right)^2 + \cdots + \left(\frac{3}{16}\right)^{k-1}\right]$

$I_c = n^2 \cdot \left[\frac{1}{1 - 3/16}\right] \implies \frac{16}{13}n^2$

Total Cost $= L_c + I_c \implies n^{\log_4 3} + \frac{16}{13}n^2$

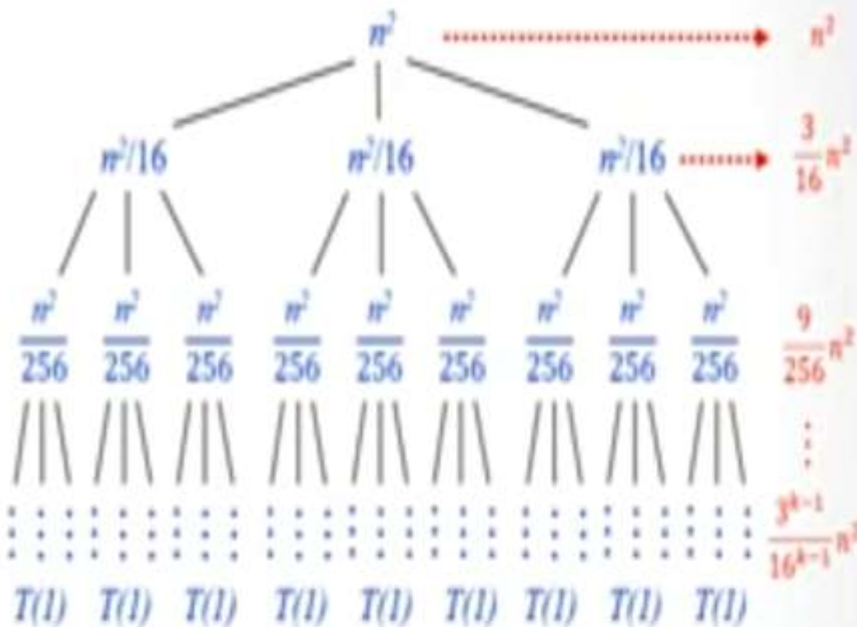
Recursion Tree Method: Example 3

$$T(n) = 3T\left(\frac{n}{4}\right) + n^2$$

$$T\left(\frac{n}{4}\right) = 3.T\left(\frac{n}{4^2}\right) + \frac{n^2}{4^2}$$

$$T\left(\frac{n}{4^2}\right) = 3.T\left(\frac{n}{4^3}\right) + \frac{n^2}{16^2}$$

$$T\left(\frac{n}{4^k}\right) = T(1)$$



$n = 4^k$

$k = \log_4 n$

$L_c = 3^k \implies 3^{\log_4 n} \implies n^{\log_4 3}$

$$I_c = n^2 \cdot \left[\left(\frac{3}{16}\right)^0 + \left(\frac{3}{16}\right)^1 + \left(\frac{3}{16}\right)^2 + \cdots + \left(\frac{3}{16}\right)^{k-1}\right]$$

$$I_c = n^2 \cdot \left[\frac{1}{1 - {}^3/_{16}}\right] \implies \frac{16}{13}n^2$$

$$\text{Total Cost} = L_c + I_c \implies n^{\log_4 3} + \frac{16}{13}n^2$$

Hence: $T(n) \in O(n^2)$

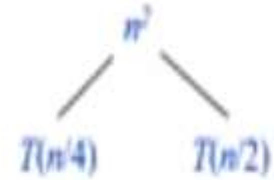
Recursion Tree Method: Example 4

$$T(n) = \begin{cases} 1 & n = 1 \\ T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n^2 & n > 1 \end{cases}$$

Solve the following recurrence using the Recurrence Tree Method.

Recursion Tree Method: Example 4

$$T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n^2$$

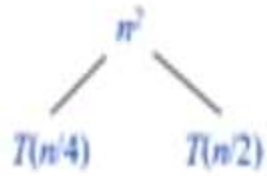


Recursion Tree Method: Example 4

$$T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n^2$$

$$T(\frac{n}{4}) = T(\frac{n}{16}) + T(\frac{n}{8}) + \frac{n^2}{16}$$

$$T(\frac{n}{2}) = T(\frac{n}{8}) + T(\frac{n}{4}) + \frac{n^2}{4}$$



Recursion Tree Method: Example 4

$$T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n^2$$

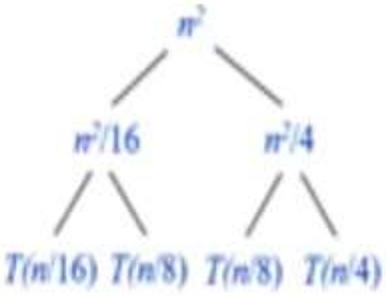
$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{16}\right) + T\left(\frac{n}{8}\right) + \frac{n^2}{16}$$

$$T\left(\frac{n}{2}\right) = T\left(\frac{n}{8}\right) + T\left(\frac{n}{4}\right) + \frac{n^2}{4}$$

$$T\left(\frac{n}{16}\right) = T\left(\frac{n}{64}\right) + T\left(\frac{n}{32}\right) + \frac{n^2}{256}$$

$$T\left(\frac{n}{8}\right) = T\left(\frac{n}{32}\right) + T\left(\frac{n}{16}\right) + \frac{n^2}{64}$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{16}\right) + T\left(\frac{n}{8}\right) + \frac{n^2}{16}$$



Recursion Tree Method: Example 4

$$T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n^2$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{16}\right) + T\left(\frac{n}{8}\right) + \frac{n^2}{16}$$

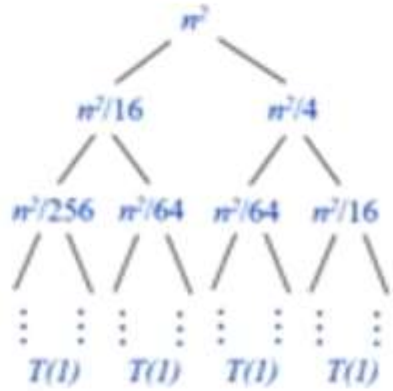
$$T\left(\frac{n}{2}\right) = T\left(\frac{n}{8}\right) + T\left(\frac{n}{4}\right) + \frac{n^2}{4}$$

$$T\left(\frac{n}{16}\right) = T\left(\frac{n}{64}\right) + T\left(\frac{n}{32}\right) + \frac{n^2}{256}$$

$$T\left(\frac{n}{8}\right) = T\left(\frac{n}{32}\right) + T\left(\frac{n}{16}\right) + \frac{n^2}{64}$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{16}\right) + T\left(\frac{n}{8}\right) + \frac{n^2}{16}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$



Recursion Tree Method: Example 4

$$T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n^2$$

$$T(\frac{n}{4}) = T(\frac{n}{16}) + T(\frac{n}{8}) + \frac{n^2}{16}$$

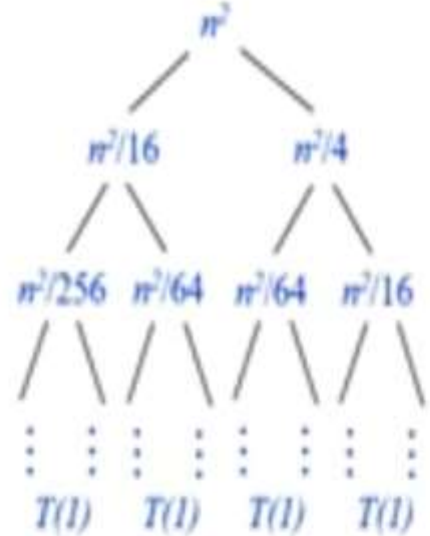
$$T(\frac{n}{2}) = T(\frac{n}{8}) + T(\frac{n}{4}) + \frac{n^2}{4}$$

$$T(\frac{n}{16}) = T(\frac{n}{64}) + T(\frac{n}{32}) + \frac{n^2}{256}$$

$$T(\frac{n}{8}) = T(\frac{n}{32}) + T(\frac{n}{16}) + \frac{n^2}{64}$$

$$T(\frac{n}{4}) = T(\frac{n}{16}) + T(\frac{n}{8}) + \frac{n^2}{16}$$

$$T(\frac{n}{2^k}) = T(1)$$



Recursion Tree Method: Example 4

$$T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n^2$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{16}\right) + T\left(\frac{n}{8}\right) + \frac{n^2}{16}$$

$$T\left(\frac{n}{2}\right) = T\left(\frac{n}{8}\right) + T\left(\frac{n}{4}\right) + \frac{n^2}{4}$$

$$T\left(\frac{n}{16}\right) = T\left(\frac{n}{64}\right) + T\left(\frac{n}{32}\right) + \frac{n^2}{256}$$

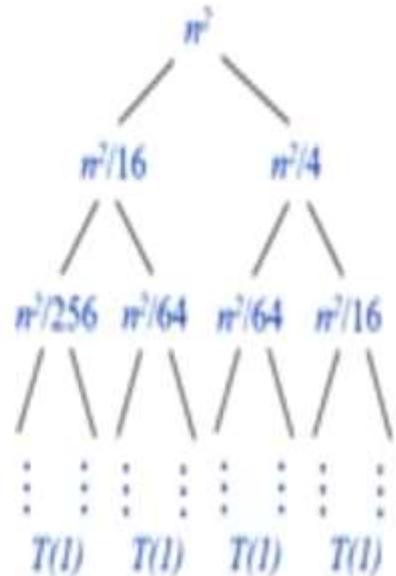
$$T\left(\frac{n}{8}\right) = T\left(\frac{n}{32}\right) + T\left(\frac{n}{16}\right) + \frac{n^2}{64}$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{16}\right) + T\left(\frac{n}{8}\right) + \frac{n^2}{16}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$

$$n = 2^k \implies k = \lg n$$

$$L_c = 2^k \implies 2^{\lg n} \implies n^{\lg 2} \implies n$$



Total Cost $= L_c + I_c$

Recursion Tree Method: Example 4

$$T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n^2$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{16}\right) + T\left(\frac{n}{8}\right) + \frac{n^2}{16}$$

$$T\left(\frac{n}{2}\right) = T\left(\frac{n}{8}\right) + T\left(\frac{n}{4}\right) + \frac{n^2}{4}$$

$$T\left(\frac{n}{16}\right) = T\left(\frac{n}{64}\right) + T\left(\frac{n}{32}\right) + \frac{n^2}{256}$$

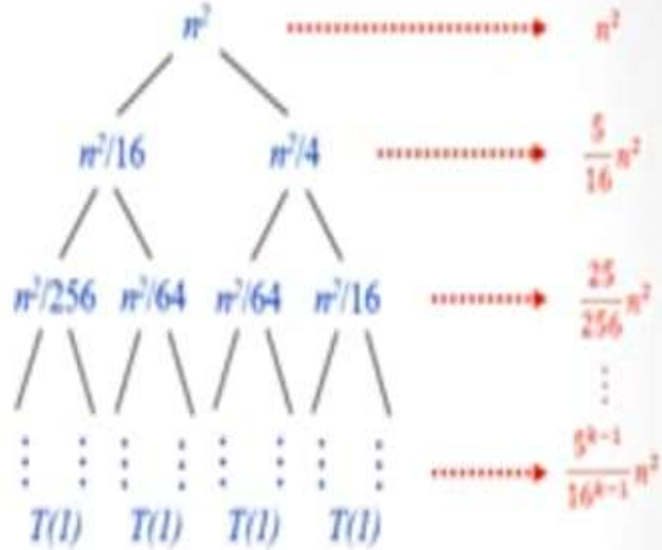
$$T\left(\frac{n}{8}\right) = T\left(\frac{n}{32}\right) + T\left(\frac{n}{16}\right) + \frac{n^2}{64}$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{16}\right) + T\left(\frac{n}{8}\right) + \frac{n^2}{16}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$

$$n = 2^k \Rightarrow k = \lg n$$

$$L_c = 2^k \Rightarrow 2^{\lg n} \Rightarrow n^{\lg 2} \Rightarrow n$$



$$I_c = n^2 \cdot \left[\left(\frac{5}{16}\right)^0 + \left(\frac{5}{16}\right)^1 + \left(\frac{5}{16}\right)^2 + \cdots + \left(\frac{5}{16}\right)^{k-1}\right]$$

$$I_c = n^2 \cdot \left[\frac{1}{1 - 5/16}\right] \Rightarrow \frac{16}{11}n^2$$

Total Cost $= L_c + I_c$

Recursion Tree Method: Example 4

$$T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n^2$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{16}\right) + T\left(\frac{n}{8}\right) + \frac{n^2}{16}$$

$$T\left(\frac{n}{2}\right) = T\left(\frac{n}{8}\right) + T\left(\frac{n}{4}\right) + \frac{n^2}{4}$$

$$T\left(\frac{n}{16}\right) = T\left(\frac{n}{64}\right) + T\left(\frac{n}{32}\right) + \frac{n^2}{256}$$

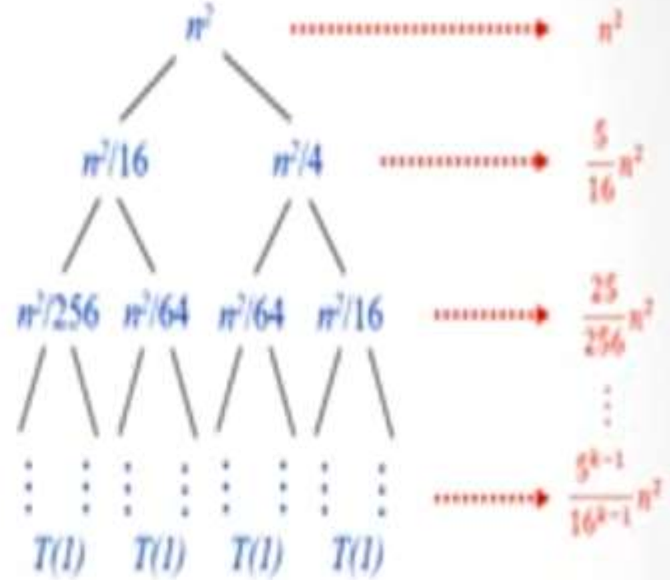
$$T\left(\frac{n}{8}\right) = T\left(\frac{n}{32}\right) + T\left(\frac{n}{16}\right) + \frac{n^2}{64}$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{16}\right) + T\left(\frac{n}{8}\right) + \frac{n^2}{16}$$

$$T\left(\frac{n}{2^k}\right) = T(1)$$

$$n = 2^k \Rightarrow k = \lg n$$

$$L_c = 2^k \Rightarrow 2^{\lg n} \Rightarrow n^{\lg 2} \Rightarrow n$$



$$I_c = n^2 \cdot \left[\left(\frac{5}{16}\right)^0 + \left(\frac{5}{16}\right)^1 + \left(\frac{5}{16}\right)^2 + \cdots + \left(\frac{5}{16}\right)^{k-1}\right]$$

$$I_c = n^2 \cdot \left[\frac{1}{1 - {}^5/_{16}}\right] \Rightarrow \frac{16}{11}n^2$$

$$\text{Total Cost} = L_c + I_c \Rightarrow n + \frac{16}{11}n^2$$ *Hence:* $T(n) \in O(n^2)$

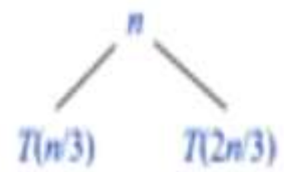
Recursion Tree Method: Example 5

$$T(n) = \begin{cases} 1 & n = 1 \\ T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n & n > 1 \end{cases}$$

Solve the following recurrence using the Recurrence Tree Method.

Recursion Tree Method: Example 5

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$

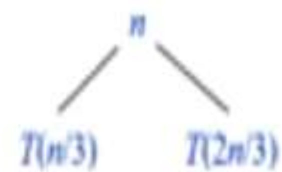


Recursion Tree Method: Example 5

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$

$$T\left(\frac{n}{3}\right) = T\left(\frac{n}{9}\right) + T\left(\frac{2n}{9}\right) + \frac{n}{3}$$

$$T\left(\frac{2n}{3}\right) = T\left(\frac{2n}{9}\right) + T\left(\frac{4n}{9}\right) + \frac{2n}{3}$$



Recursion Tree Method: Example 5

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$

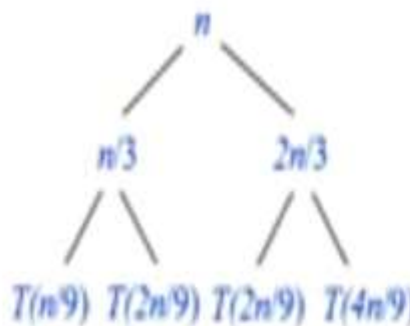
$$T\left(\frac{n}{3}\right) = T\left(\frac{n}{9}\right) + T\left(\frac{2n}{9}\right) + \frac{n}{3}$$

$$T\left(\frac{2n}{3}\right) = T\left(\frac{2n}{9}\right) + T\left(\frac{4n}{9}\right) + \frac{2n}{3}$$

$$T\left(\frac{n}{9}\right) = T\left(\frac{n}{27}\right) + T\left(\frac{2n}{27}\right) + \frac{n}{9}$$

$$T\left(\frac{2n}{9}\right) = T\left(\frac{2n}{27}\right) + T\left(\frac{4n}{27}\right) + \frac{2n}{9}$$

$$T\left(\frac{4n}{9}\right) = T\left(\frac{4n}{27}\right) + T\left(\frac{8n}{27}\right) + \frac{4n}{9}$$



Recursion Tree Method: Example 5

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$

$$T\left(\frac{n}{3}\right) = T\left(\frac{n}{9}\right) + T\left(\frac{2n}{9}\right) + \frac{n}{3}$$

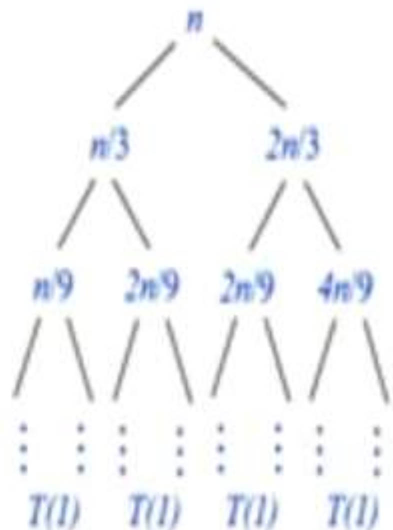
$$T\left(\frac{2n}{3}\right) = T\left(\frac{2n}{9}\right) + T\left(\frac{4n}{9}\right) + \frac{2n}{3}$$

$$T\left(\frac{n}{9}\right) = T\left(\frac{n}{27}\right) + T\left(\frac{2n}{27}\right) + \frac{n}{9}$$

$$T\left(\frac{2n}{9}\right) = T\left(\frac{2n}{27}\right) + T\left(\frac{4n}{27}\right) + \frac{2n}{9}$$

$$T\left(\frac{4n}{9}\right) = T\left(\frac{4n}{27}\right) + T\left(\frac{8n}{27}\right) + \frac{4n}{9}$$

$$T\left(\frac{2^k}{3^k}n\right) = T(1)$$



Recursion Tree Method: Example 5

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$

$$T\left(\frac{n}{3}\right) = T\left(\frac{n}{9}\right) + T\left(\frac{2n}{9}\right) + \frac{n}{3}$$

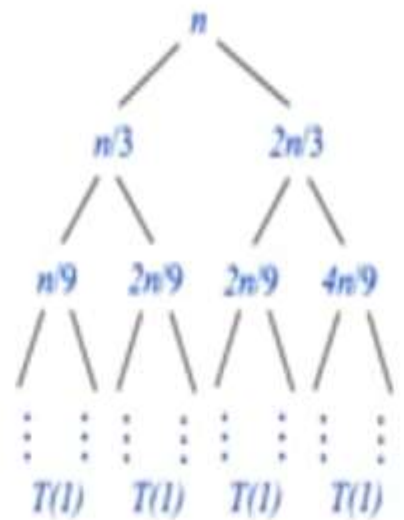
$$T\left(\frac{2n}{3}\right) = T\left(\frac{2n}{9}\right) + T\left(\frac{4n}{9}\right) + \frac{2n}{3}$$

$$T\left(\frac{n}{9}\right) = T\left(\frac{n}{27}\right) + T\left(\frac{2n}{27}\right) + \frac{n}{9}$$

$$T\left(\frac{2n}{9}\right) = T\left(\frac{2n}{27}\right) + T\left(\frac{4n}{27}\right) + \frac{2n}{9}$$

$$T\left(\frac{4n}{9}\right) = T\left(\frac{4n}{27}\right) + T\left(\frac{8n}{27}\right) + \frac{4n}{9}$$

$$T\left(\frac{2^k}{3^k}n\right) = T(1)$$



Recursion Tree Method: Example 5

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$

$$T(\frac{n}{3}) = T(\frac{n}{9}) + T(\frac{2n}{9}) + \frac{n}{3}$$

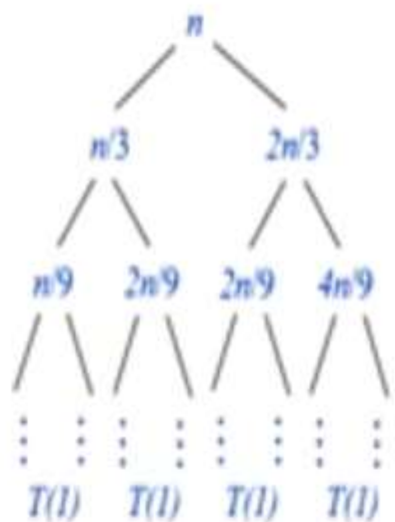
$$T(\frac{2n}{3}) = T(\frac{2n}{9}) + T(\frac{4n}{9}) + \frac{2n}{3}$$

$$T(\frac{n}{9}) = T(\frac{n}{27}) + T(\frac{2n}{27}) + \frac{n}{9}$$

$$T(\frac{2n}{9}) = T(\frac{2n}{27}) + T(\frac{4n}{27}) + \frac{2n}{9}$$

$$T(\frac{4n}{9}) = T(\frac{4n}{27}) + T(\frac{8n}{27}) + \frac{4n}{9}$$

$$T(\frac{2^k}{3^k}n) = T(1) \implies n = \frac{3^k}{2^k} \implies k = \log_{3/2} n$$



Total Cost $= L_c + I_c$

Recursion Tree Method: Example 5

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$

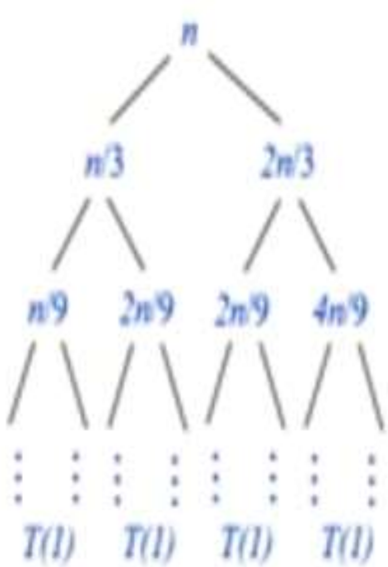
$$T\left(\frac{n}{3}\right) = T\left(\frac{n}{9}\right) + T\left(\frac{2n}{9}\right) + \frac{n}{3}$$

$$T\left(\frac{2n}{3}\right) = T\left(\frac{2n}{9}\right) + T\left(\frac{4n}{9}\right) + \frac{2n}{3}$$

$$T\left(\frac{n}{9}\right) = T\left(\frac{n}{27}\right) + T\left(\frac{2n}{27}\right) + \frac{n}{9}$$

$$T\left(\frac{2n}{9}\right) = T\left(\frac{2n}{27}\right) + T\left(\frac{4n}{27}\right) + \frac{2n}{9}$$

$$T\left(\frac{4n}{9}\right) = T\left(\frac{4n}{27}\right) + T\left(\frac{8n}{27}\right) + \frac{4n}{9}$$



$$T\left(\frac{2^k}{3^k}n\right) = T(1) \implies n = \frac{3^k}{2^k} \implies k = \log_{3/2} n$$

$$L_c = 2^k \implies 2^{\log_{3/2} n} \implies n^{\log_{3/2} 2}$$

Total Cost $= L_c + I_c$

Recursion Tree Method: Example 5

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$

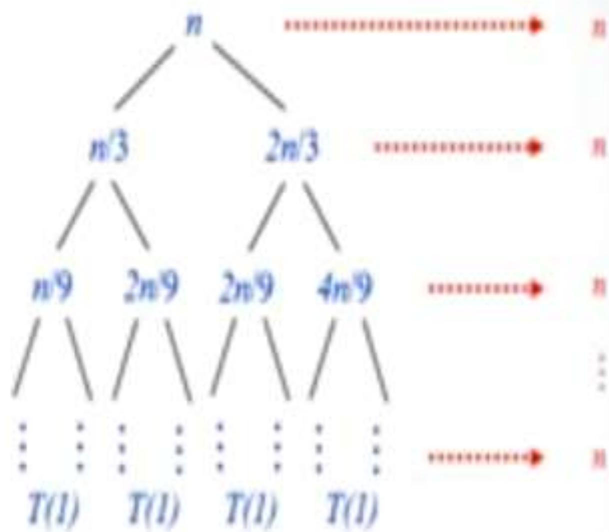
$$T\left(\frac{n}{3}\right) = T\left(\frac{n}{9}\right) + T\left(\frac{2n}{9}\right) + \frac{n}{3}$$

$$T\left(\frac{2n}{3}\right) = T\left(\frac{2n}{9}\right) + T\left(\frac{4n}{9}\right) + \frac{2n}{3}$$

$$T\left(\frac{n}{9}\right) = T\left(\frac{n}{27}\right) + T\left(\frac{2n}{27}\right) + \frac{n}{9}$$

$$T\left(\frac{2n}{9}\right) = T\left(\frac{2n}{27}\right) + T\left(\frac{4n}{27}\right) + \frac{2n}{9}$$

$$T\left(\frac{4n}{9}\right) = T\left(\frac{4n}{27}\right) + T\left(\frac{8n}{27}\right) + \frac{4n}{9}$$



$$l_c = n \cdot k$$

$$l_c = n \log_{3/2} n$$

$$T\left(\frac{2^k}{3^k} n\right) = T(1) \implies n = \frac{3^k}{2^k} \implies k = \log_{3/2} n$$

$$L_c = 2^k \implies 2^{\log_{3/2} n} \implies n^{\log_{3/2} 2}$$

$$\text{Total Cost} = L_c + l_c \implies n^{\log_{3/2} 2} + n \log_{3/2} n$$

Recursion Tree Method: Example 5

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$

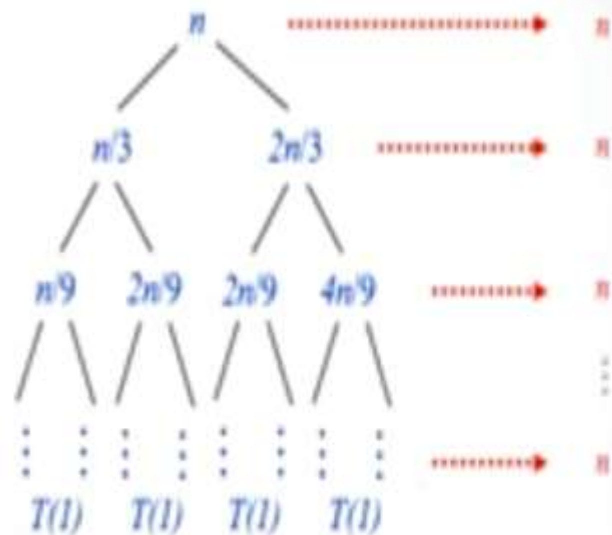
$$T\left(\frac{n}{3}\right) = T\left(\frac{n}{9}\right) + T\left(\frac{2n}{9}\right) + \frac{n}{3}$$

$$T\left(\frac{2n}{3}\right) = T\left(\frac{2n}{9}\right) + T\left(\frac{4n}{9}\right) + \frac{2n}{3}$$

$$T\left(\frac{n}{9}\right) = T\left(\frac{n}{27}\right) + T\left(\frac{2n}{27}\right) + \frac{n}{9}$$

$$T\left(\frac{2n}{9}\right) = T\left(\frac{2n}{27}\right) + T\left(\frac{4n}{27}\right) + \frac{2n}{9}$$

$$T\left(\frac{4n}{9}\right) = T\left(\frac{4n}{27}\right) + T\left(\frac{8n}{27}\right) + \frac{4n}{9}$$



$$l_c = n \cdot k$$

$$l_c = n \log_{3/2} n$$

$$T\left(\frac{2^k}{3^k} n\right) = T(1) \implies n = \frac{3^k}{2^k} \implies k = \log_{3/2} n$$

$$L_c = 2^k \implies 2^{\log_{3/2} n} \implies n^{\log_{3/2} 2}$$

$$\text{Total Cost} = L_c + l_c \implies n^{\log_{3/2} 2} + n \log_{3/2} n$$

Recursion Tree Method: Example 5

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$

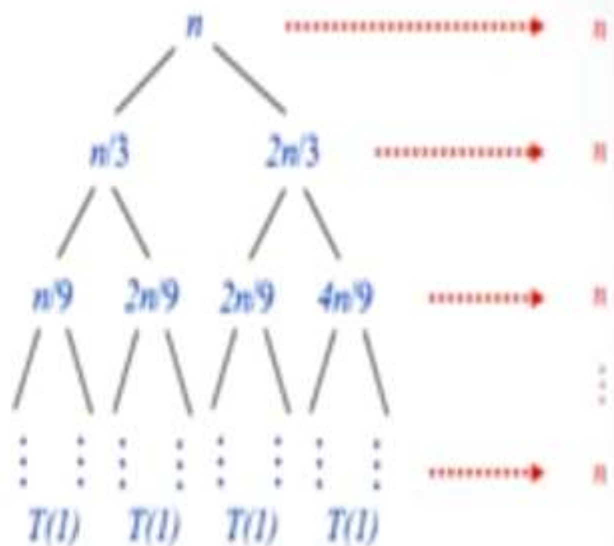
$$T\left(\frac{n}{3}\right) = T\left(\frac{n}{9}\right) + T\left(\frac{2n}{9}\right) + \frac{n}{3}$$

$$T\left(\frac{2n}{3}\right) = T\left(\frac{2n}{9}\right) + T\left(\frac{4n}{9}\right) + \frac{2n}{3}$$

$$T\left(\frac{n}{9}\right) = T\left(\frac{n}{27}\right) + T\left(\frac{2n}{27}\right) + \frac{n}{9}$$

$$T\left(\frac{2n}{9}\right) = T\left(\frac{2n}{27}\right) + T\left(\frac{4n}{27}\right) + \frac{2n}{9}$$

$$T\left(\frac{4n}{9}\right) = T\left(\frac{4n}{27}\right) + T\left(\frac{8n}{27}\right) + \frac{4n}{9}$$



$$I_c = n \cdot k$$

$$I_c = n \log_{3/2} n$$

$$T\left(\frac{2^k}{3^k} n\right) = T(1) \implies n = \frac{3^k}{2^k} \implies k = \log_{3/2} n$$

$$L_c = 2^k \implies 2^{\log_{3/2} n} \implies n^{\log_{3/2} 2}$$

$$\text{Total Cost} = L_c + I_c \implies n^{\log_{3/2} 2} + n \log_{3/2} n$$

Recursion Tree Method: Example 5

$$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$$

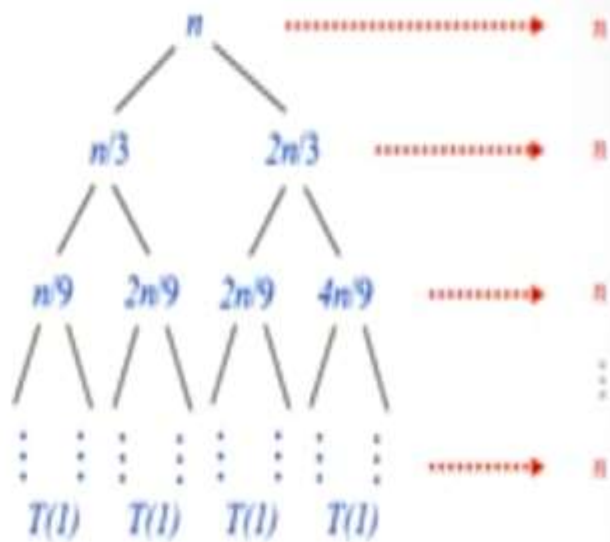
$$T\left(\frac{n}{3}\right) = T\left(\frac{n}{9}\right) + T\left(\frac{2n}{9}\right) + \frac{n}{3}$$

$$T\left(\frac{2n}{3}\right) = T\left(\frac{2n}{9}\right) + T\left(\frac{4n}{9}\right) + \frac{2n}{3}$$

$$T\left(\frac{n}{9}\right) = T\left(\frac{n}{27}\right) + T\left(\frac{2n}{27}\right) + \frac{n}{9}$$

$$T\left(\frac{2n}{9}\right) = T\left(\frac{2n}{27}\right) + T\left(\frac{4n}{27}\right) + \frac{2n}{9}$$

$$T\left(\frac{4n}{9}\right) = T\left(\frac{4n}{27}\right) + T\left(\frac{8n}{27}\right) + \frac{4n}{9}$$



$$l_c = n \cdot k$$

$$l_c = n \log_{3/2} n$$

$$T\left(\frac{2^k}{3^k} n\right) = T(1) \Rightarrow n = \frac{3^k}{2^k} \Rightarrow k = \log_{3/2} n$$

$$L_c = 2^k \Rightarrow 2^{\log_{3/2} n} \Rightarrow n^{\log_{3/2} 2}$$

Total Cost $= L_c + l_c \Rightarrow n^{\log_{3/2} 2} + n \log_{3/2} n$ $\quad T(n) \in O(n \lg n)$??

Recursion Tree Method: Caution Note

- **Recursion Trees** are best used to generate good guesses.
  - Verify guesses using the substitution method.
- A small amount of "sloppiness" can be tolerated.
  - Using an infinite decreasing geometric series as an upper bound.
  - Assuming "*n*" to be an exact power of 2, 3, or 4.

Recursion Tree Method: Caution Note

- **Recursion Trees** are best used to generate good guesses.
  - Verify guesses using the substitution method.
- A small amount of "sloppiness" can be tolerated.
  - Using an infinite decreasing geometric series as an upper bound.
  - Assuming "n" to be an exact power of 2, 3, or 4.
- By carefully drawing out a recursion tree and summing the costs, recursion tree method can be used as a direct proof of a solution to any recurrence.

Practice Questions

Solve the following recurrences using the Recurrence Tree Method.

$$T(n) = 4T\left(\frac{n}{2}\right) + n^2$$

$$T(n) = 2T\left(\frac{n}{3}\right) + n$$

$$T(n) = 2T(n-1) + 1$$

$$T(n) = 3T\left(\frac{n}{4}\right) + n$$

$$T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) + n$$

Master Method

Master Method is a method for solving recurrences of the form:

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

also known as "Master Recurrence", where $a \geq 1$ and $b > 1$ are constants and $f(n)$ is an asymptotically positive function.

The **Master Method** requires memorization of three cases, after which the solution of many recurrences of this form can be solved quite easily with very little work.

Master Method

- Let's solve the Master Recurrence using the Recurrence Tree Method to see what is going on.

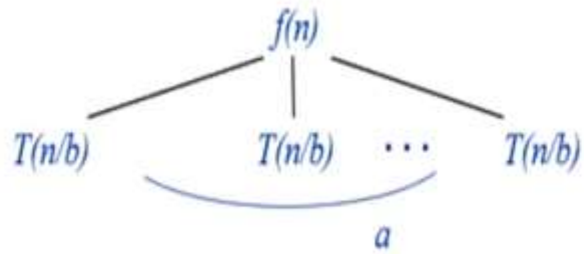
$$T(n) = \begin{cases} 1 & n = 1 \\ a\,T\left(\frac{n}{b}\right) + f(n) & n > 1, b > 1, a \geq 1 \end{cases}$$

f is asymptotically positive.

- Assumption: n is exact power of b.

Master Method

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

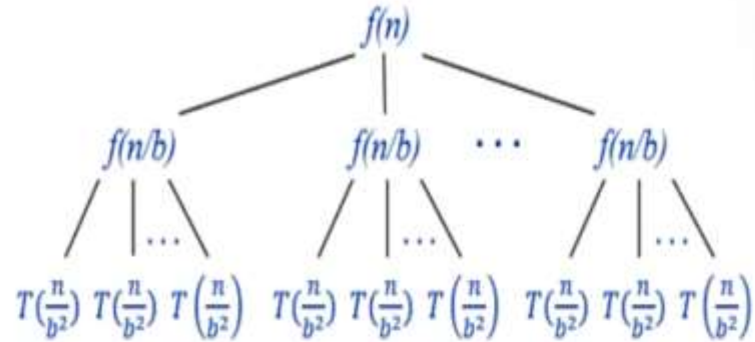


Master Method

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

$$T\left(\frac{n}{b}\right) = a.T\left(\frac{n}{b^2}\right) + f\left(\frac{n}{b}\right)$$

$$T\left(\frac{n}{b^2}\right) = a.T\left(\frac{n}{b^3}\right) + f\left(\frac{n}{b^2}\right)$$



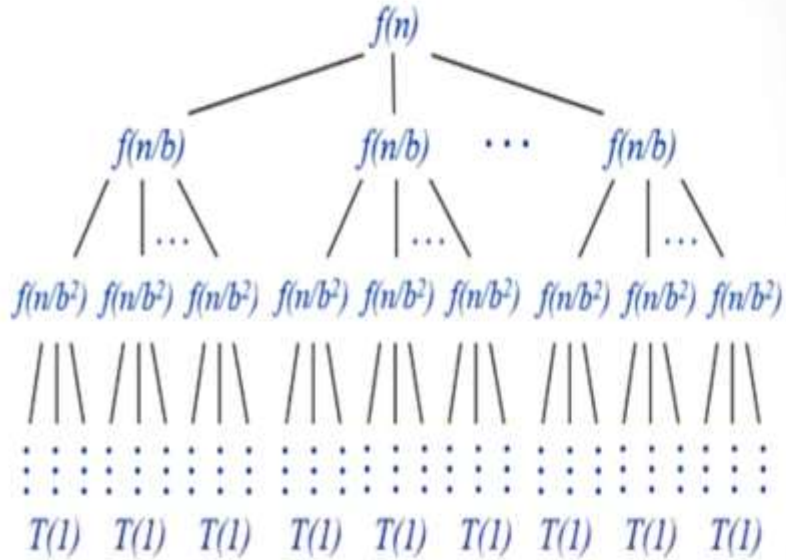
Master Method

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

$$T(\frac{n}{b}) = a.T(\frac{n}{b^2}) + f(\frac{n}{b})$$

$$T(\frac{n}{b^2}) = a.T(\frac{n}{b^3}) + f(\frac{n}{b^2})$$

$$T(\frac{n}{b^k}) = T(1)$$



Master Method

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

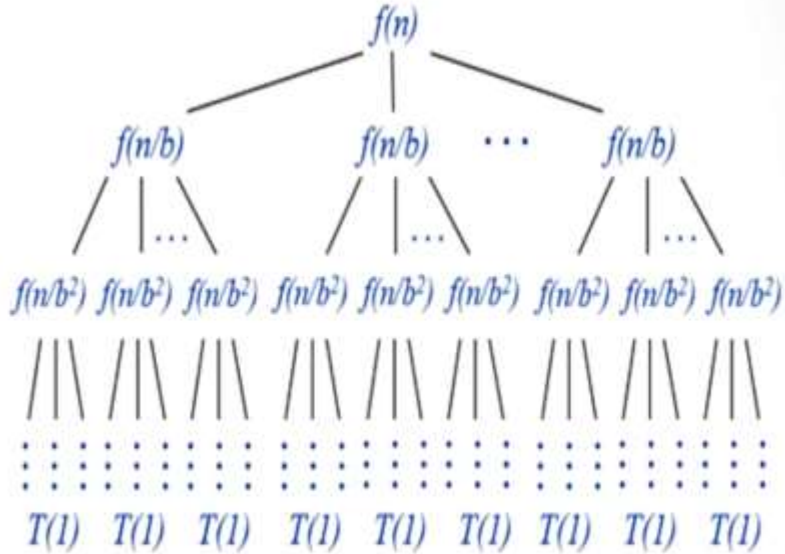
$$T(\frac{n}{b}) = a.T(\frac{n}{b^2}) + f(\frac{n}{b})$$

$$T(\frac{n}{b^2}) = a.T(\frac{n}{b^3}) + f(\frac{n}{b^2})$$

$$T(\frac{n}{b^k}) = T(1)$$

$$n = b^k$$

$$k = \log_b n$$



Total Cost = $L_c + I_c$

Master Method

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

$$T\left(\frac{n}{b}\right) = a.T\left(\frac{n}{b^2}\right) + f\left(\frac{n}{b}\right)$$

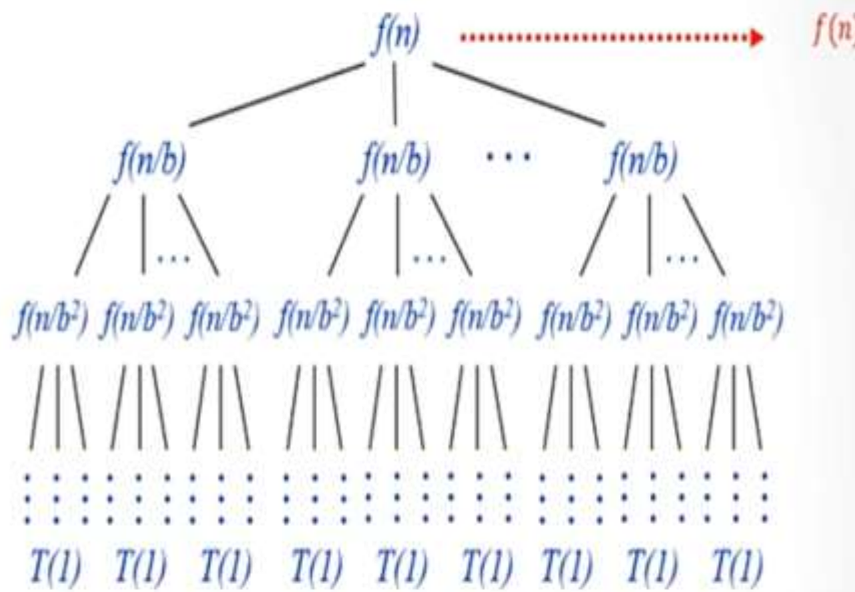
$$T\left(\frac{n}{b^2}\right) = a.T\left(\frac{n}{b^3}\right) + f\left(\frac{n}{b^2}\right)$$

$$T\left(\frac{n}{b^k}\right) = T(1)$$

$$n = b^k$$

$$k = \log_b n$$

$$L_c = a^k \implies a^{\log_b n} \implies n^{\log_b a}$$



Total Cost $= L_c + I_c$

Master Method

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

$$T\left(\frac{n}{b}\right) = a.T\left(\frac{n}{b^2}\right) + f\left(\frac{n}{b}\right)$$

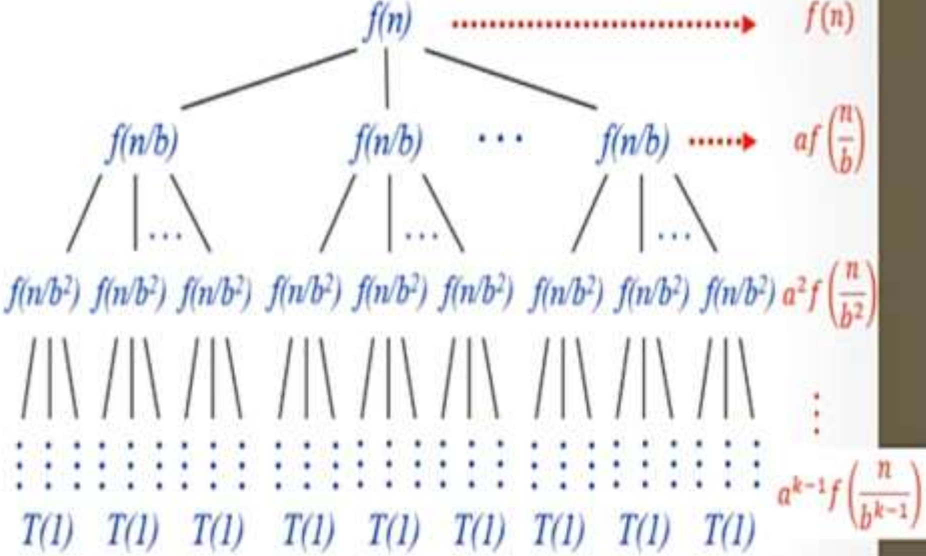
$$T\left(\frac{n}{b^2}\right) = a.T\left(\frac{n}{b^3}\right) + f\left(\frac{n}{b^2}\right)$$

$$T\left(\frac{n}{b^k}\right) = T(1)$$

$$n = b^k$$

$$k = \log_b n$$

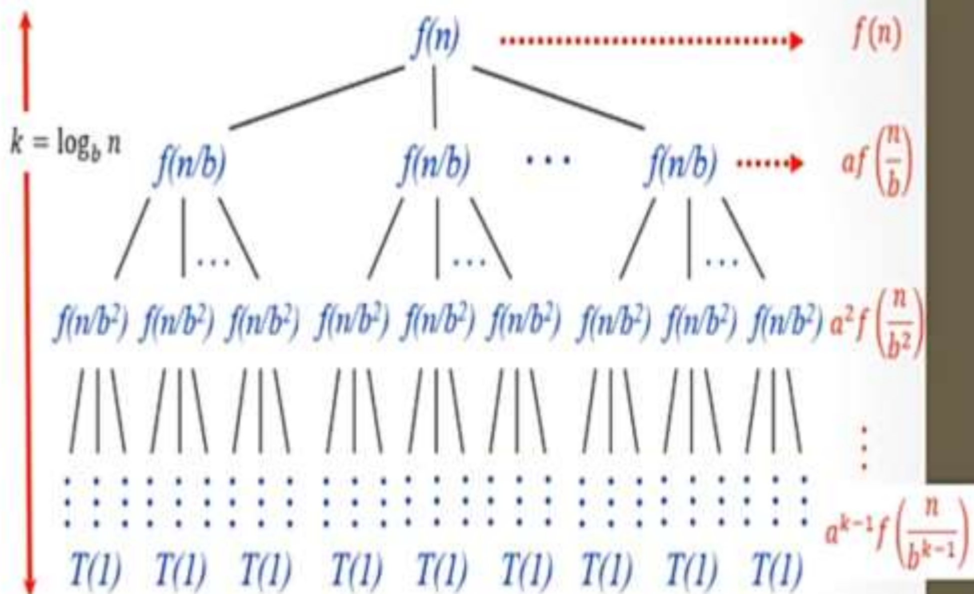
$$L_c = a^k \implies a^{\log_b n} \implies n^{\log_b a}$$



$$\text{Total Cost} = L_c + I_c \implies n^{\log_b a} + \sum_{k=0}^{\log_b n - 1} a^k f\left(\frac{n}{b^k}\right)$$

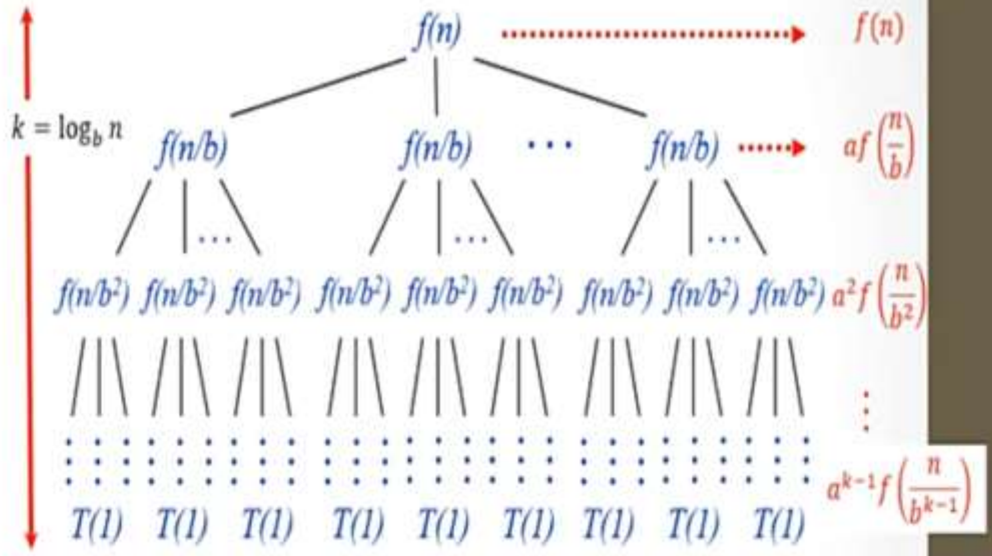
Master Method

Idea: Compare $f(n)$ with $n^{\log_b a}$.



Master Method

Idea: Compare $f(n)$ with $n^{\log_b a}$.

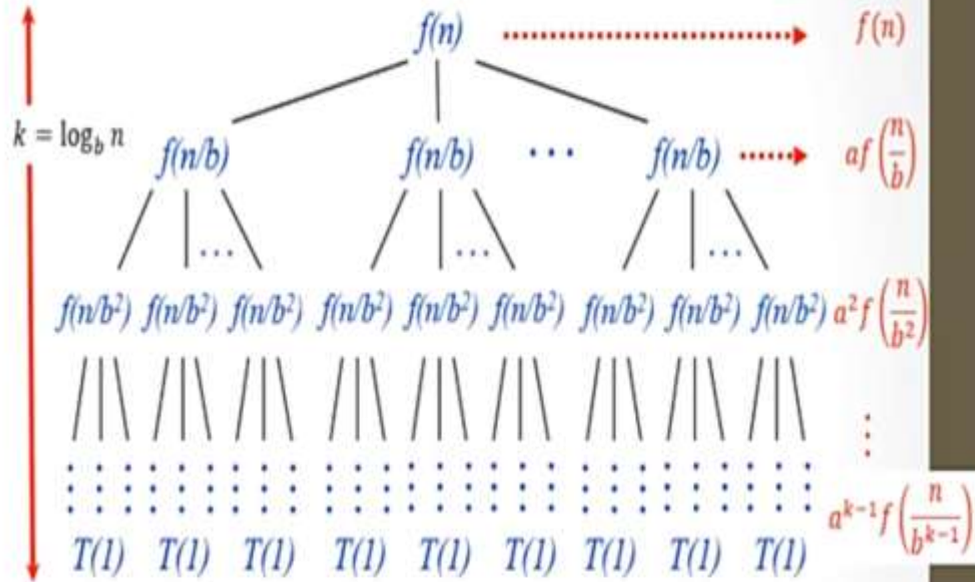


CASE 1:
Cost increases geometrically from the root to the leaves. $n^{\log_b a}$ is asymptotically larger in growth than $f(n)$ by a polynomial factor n^{ε}.

Master Method

Idea: Compare $f(n)$ with $n^{\log_b a}$.

1. $T(n) = \Theta\left(n^{\log_b a}\right)$



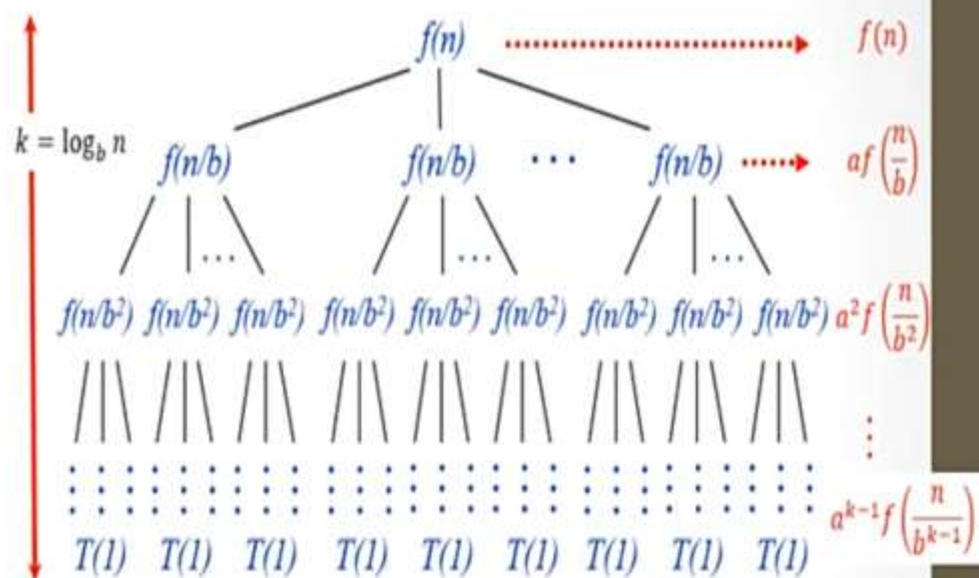
CASE 1:
Cost increases geometrically from the root to the leaves.
$n^{\log_b a}$ is asymptotically larger in growth than $f(n)$ by a polynomial factor n^{ε}.

CASE 2:
Cost is approximately the same on each of the **$\log_b n$** levels.
The growth of $n^{\log_b a}$ is asymptotically equal to $f(n)$.

Master Method

Idea: Compare $f(n)$ with $n^{\log_b a}$.

1. $T(n) = \Theta\left(n^{\log_b a}\right)$
2. $T(n) = \Theta\left(n^{\log_b a} \log_b n\right)$



CASE 1:
Cost increases geometrically from the root to the leaves.
$n^{\log_b a}$ is asymptotically larger in growth than $f(n)$ by a polynomial factor n^{ε}.

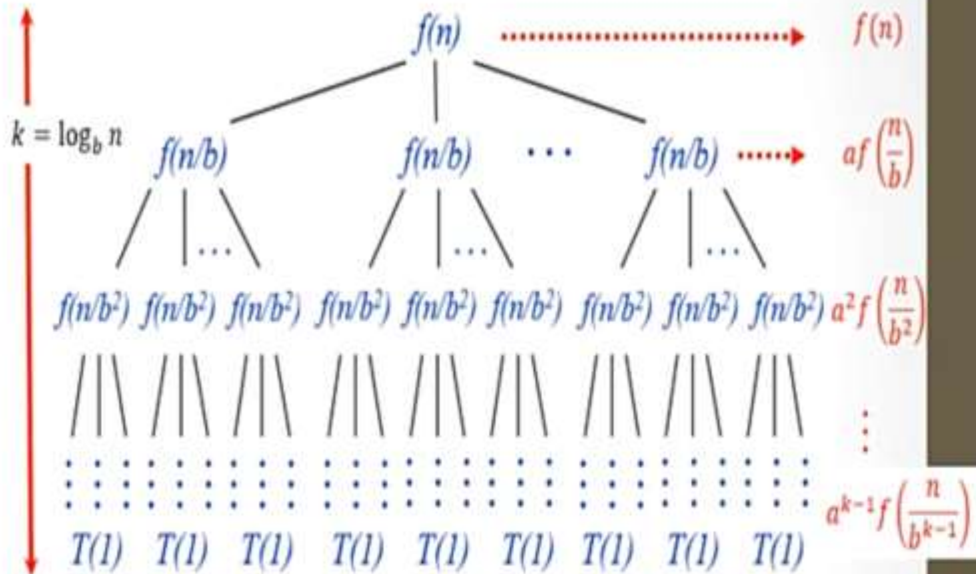
CASE 2:
Cost is approximately the same on each of the **$\log_b n$** levels.
The growth of $n^{\log_b a}$ is asymptotically equal to $f(n)$.

CASE 3:
Cost decreases geometrically from the root to the leaves.
$n^{\log_b a}$ is asymptotically smaller in growth than $f(n)$ by a polynomial factor n^{ε}.

Master Method

Idea: Compare $f(n)$ with $n^{\log_b a}$.

1. $T(n) = \Theta\left(n^{\log_b a}\right)$
2. $T(n) = \Theta\left(n^{\log_b a} \log_b n\right)$
3. $T(n) = \Theta(f(n))$



CASE 1:
Cost increases geometrically from the root to the leaves.
$n^{\log_b a}$ is asymptotically larger in growth than $f(n)$ by a polynomial factor n^{ε}.

CASE 2:
Cost is approximately the same on each of the $\log_b n$ levels.
The growth of $n^{\log_b a}$ is asymptotically equal to $f(n)$.

CASE 3:
Cost decreases geometrically from the root to the leaves.
$n^{\log_b a}$ is asymptotically smaller in growth than $f(n)$ by a polynomial factor n^{ε}.

The Master Theorem

The **Master Method** depends on the following Theorem:

Theorem: Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence:

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

Then $T(n)$ can be bounded asymptotically as follows.

1. If $f(n) = O(n^{\log_b a-\varepsilon})$ for some constant $\varepsilon > 0$, then $T(n) = \Theta\left(n^{\log_b a}\right)$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta\left(n^{\log_b a} \lg n\right)$.
3. If $f(n) = \Omega(n^{\log_b a+\varepsilon})$ for some constant $\varepsilon > 0$, and if $a \cdot f\left(\frac{n}{b}\right) \leq cf(n)$ for some constant c < 1 and all sufficiently large n, then $T(n) = \Theta(f(n))$.

The Master Theorem

The **Master Method** depends on the following Theorem:

Theorem: Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence:

$$T(n) = a\,T\left(\frac{n}{b}\right) + f(n)$$

Then $T(n)$ can be bounded asymptotically as follows.

1. If $f(n) = O(n^{\log_b a-\varepsilon})$ for some constant $\varepsilon > 0$, then $T(n) = \Theta\left(n^{\log_b a}\right)$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta\left(n^{\log_b a} \lg n\right)$.
3. If $f(n) = \Omega(n^{\log_b a+\varepsilon})$ for some constant $\varepsilon > 0$, and if $a \cdot f\left(\frac{n}{b}\right) \leq cf(n)$ for some constant c < 1 and all sufficiently large n, then $T(n) = \Theta(f(n))$.

regularity condition

Master Method: Case 1→ $n^{\log_b a} > f(n)$

$\left(\frac{n^{\log_b a}}{f(n)}\right) = \Omega(n^{\varepsilon})$

Master Method: Case 1 → $n^{\log_b a} > f(n)$

$$\left(\frac{n^{\log_b a}}{f(n)}\right) = \Omega(n^{\varepsilon}) \Rightarrow \left(\frac{n^{\log_b a}}{f(n)}\right) \geq cn^{\varepsilon} \Rightarrow f(n) \leq \left(\frac{n^{\log_b a}}{cn^{\varepsilon}}\right) \Rightarrow f(n) \leq cn^{\log_b a-\varepsilon} \Rightarrow f(n) = O(n^{\log_b a-\varepsilon})$$

$$T(n) = n^{\log_b a} + \sum_{k=0}^{\log_b n-1} a^k f\left(\frac{n}{b^k}\right)$$

Suppose $f(n) = n^{\delta}$, where $\delta < \log_b a$ or $b^{\delta} < a$. Then:

$$T(n) = n^{\log_b a} + \sum_{k=0}^{\log_b n-1} \left(\frac{a}{b^{\delta}}\right)^k n^{\delta} \Rightarrow n^{\log_b a} + n^{\delta}\left[\frac{\left(\frac{a}{b^{\delta}}\right)^{\log_b n} - 1}{\frac{a}{b^{\delta}} - 1}\right] \Rightarrow n^{\log_b a} + n^{\delta}\left[\frac{\frac{n^{\log_b a}}{n^{\delta}} - 1}{c}\right]$$

$$\Rightarrow n^{\log_b a} + \frac{n^{\log_b a} - n^{\delta}}{c} \Rightarrow \Theta\left(n^{\log_b a}\right)$$

Case 1: If $f(n) = O(n^{\log_b a-\varepsilon})$ for some constant $\varepsilon > 0$, then $T(n) = \Theta\left(n^{\log_b a}\right)$.

Master Method: Case 1 → $n^{\log_b a} > f(n)$

$$\left(\frac{n^{\log_b a}}{f(n)}\right) = \Omega(n^{\varepsilon}) \Rightarrow \left(\frac{n^{\log_b a}}{f(n)}\right) \geq cn^{\varepsilon} \Rightarrow f(n) \leq \left(\frac{n^{\log_b a}}{cn^{\varepsilon}}\right) \Rightarrow f(n) \leq cn^{\log_b a-\varepsilon} \Rightarrow f(n) = O(n^{\log_b a-\varepsilon})$$

$$T(n) = n^{\log_b a} + \sum_{k=0}^{\log_b n-1} a^k f\left(\frac{n}{b^k}\right)$$

Suppose $f(n) = n^{\delta}$, where $\delta < \log_b a$ or $b^{\delta} < a$. Then:

$$T(n) = n^{\log_b a} + \sum_{k=0}^{\log_b n-1} \left(\frac{a}{b^{\delta}}\right)^k n^{\delta} \Rightarrow n^{\log_b a} + n^{\delta}\left[\frac{\left(\frac{a}{b^{\delta}}\right)^{\log_b n} - 1}{\frac{a}{b^{\delta}} - 1}\right] \Rightarrow n^{\log_b a} + n^{\delta}\left[\frac{\frac{n^{\log_b a}}{n^{\delta}} - 1}{c}\right]$$

$$\Rightarrow n^{\log_b a} + \frac{n^{\log_b a} - n^{\delta}}{c} \Rightarrow \Theta\left(n^{\log_b a}\right)$$

Case 1: If $f(n) = O(n^{\log_b a-\varepsilon})$ for some constant $\varepsilon > 0$, then $T(n) = \Theta\left(n^{\log_b a}\right)$.

Master Method: Case 2 → $n^{\log_b a} = f(n)$

$f(n) = \Theta(n^{\log_b a})$

$$T(n) = n^{\log_b a} + \sum_{k=0}^{\log_b n - 1} a^k f\left(\frac{n}{b^k}\right)$$

Suppose $f(n) = n^{\delta}$, where $\delta = \log_b a$ or $b^{\delta} = a$. Then:

$$T(n) = n^{\log_b a} + \sum_{k=0}^{\log_b n - 1} \left(\frac{a}{b^{\delta}}\right)^k n^{\delta} \Rightarrow n^{\log_b a} + \sum_{k=0}^{\log_b n - 1} n^{\delta} \Rightarrow n^{\log_b a} + n^{\delta} \log_b n$$

$$\Rightarrow n^{\log_b a} + n^{\log_b a} \log_b n$$

$$\Rightarrow n^{\log_b a} + c \cdot n^{\log_b a} \lg n$$

$$\log_b n = \frac{\log_2 n}{\log_2 b}$$

$$\Rightarrow \Theta\left(n^{\log_b a} \lg n\right)$$

Case 2: If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta\left(n^{\log_b a} \lg n\right)$.

Master Method: Case 3→ $n^{\log_b a} < f(n)$

$$\left(\frac{f(n)}{n^{\log_b a}}\right) = \Omega(n^\varepsilon) \Rightarrow \left(\frac{f(n)}{n^{\log_b a}}\right) \geq cn^\varepsilon \Rightarrow f(n) \geq cn^\varepsilon \cdot n^{\log_b a} \Rightarrow f(n) \geq cn^{\log_b a+\varepsilon} \Rightarrow f(n) = \Omega(n^{\log_b a+\varepsilon})$$

$$T(n) = n^{\log_b a} + \sum_{k=0}^{\log_b n-1} a^k f\left(\frac{n}{b^k}\right)$$

Suppose $f(n) = n^\delta$, where $\delta > \log_b a$ or $b^\delta > a$. Then:

$$T(n) = n^{\log_b a} + \sum_{k=0}^{\log_b n-1} \left(\frac{a}{b^\delta}\right)^k n^\delta \Rightarrow n^{\log_b a} + n^\delta\left[\frac{1}{1-\frac{a}{b^\delta}}\right] \Rightarrow n^{\log_b a} + c \cdot n^\delta \Rightarrow \Theta(f(n))$$

Case 3: If $f(n) = \Omega(n^{\log_b a+\varepsilon})$ for some constant $\varepsilon > 0$, and if $a \cdot f\left(\frac{n}{b}\right) \leq cf(n)$ for some constant c < 1 and all sufficiently large n, then $T(n) = \Theta(f(n))$.

Master Method

- Important to note that the three cases do not cover all the possibilities.
  - Gap between cases 1 and 2 when $f(n)$ is smaller than $n^{\log_b a}$ but not polynomially smaller.
  - Gap between cases 2 and 3 when $f(n)$ is larger than $n^{\log_b a}$ but not polynomially larger.

- If $f(n)$ falls into one of these gaps, or if the regularity condition in case 3 fails to hold, the master method cannot be used to solve the recurrence.

Master Method: Example 1

$$T(n) = 2\,T\left(\frac{n}{2}\right) + n$$

Master Method: Example 1

$$T(n) = 2\,T\left(\frac{n}{2}\right) + n$$

$a = 2$

$b = 2$

$f(n) = n$

$n^{\log_b a} \Rightarrow n^{\log_2 2} \Rightarrow n$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) = n^{\log_b a}$ so case 2 is applied. $\left[f(n) = \Theta(n^{\log_b a})\right]$

$\Rightarrow T(n) = \Theta\left(n^{\log_b a} \lg n\right)$

$= \Theta\left(n^{\log_2 2} \lg n\right)$

$= \Theta(n \lg n)$

Hence: $T(n) = \Theta(n \lg n)$

Master Method: Example 2

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

$a = 2$

$b = 2$

$f(n) = n^2$

$n^{\log_b a} \Rightarrow n^{\log_2 2} \Rightarrow n$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) > n^{\log_b a}$ so case 3 is applied. $\left[f(n) = \Omega(n^{\log_b a+\varepsilon})\right]$

$\Rightarrow T(n) = \Theta(f(n))$

$= \Theta(n^2)$

Hence: $T(n) = \Theta(n^2)$

Verify Regularity Condition:

$$a \cdot f\left(\frac{n}{b}\right) \le cf(n)$$

$$2 \cdot f\left(\frac{n}{2}\right) \le cn^2$$

$$2 \cdot \frac{n^2}{4} \le cn^2$$

$$\frac{1}{2} \le c$$

Master Method: Example 3

$$T(n) = 9\,T\left(\frac{n}{3}\right) + n$$

$a = 9$

$b = 3$

$f(n) = n$

$n^{\log_b a} \Rightarrow n^{\log_3 9} \Rightarrow n^2$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) < n^{\log_b a}$ so case 1 is applied. $\left[f(n) = O(n^{\log_b a-\varepsilon})\right]$

$\Rightarrow T(n) = \Theta\left(n^{\log_b a}\right)$

$= \Theta\left(n^{\log_3 9}\right)$

$= \Theta(n^2)$

Hence: $T(n) = \Theta(n^2)$

Master Method: Example 4

$$T(n) = T\left(\frac{n}{2}\right) + 1$$

16

Master Method: Example 4

$$T(n) = T\left(\frac{n}{2}\right) + 1$$

$a = 1$

$b = 2$

$f(n) = n^0$

$n^{\log_b a} \Longrightarrow n^{\log_2 1} \Longrightarrow n^0$

Compare $f(n)$ and $n^{\log_b a}$:

$\Longrightarrow f(n) = n^{\log_b a}$ so case 2 is applied. $\left[f(n) = \Theta(n^{\log_b a})\right]$

$$\Longrightarrow T(n) = \Theta\left(n^{\log_b a} \lg n\right)$$

$$= \Theta\left(n^{\log_2 1} \lg n\right)$$

$$= \Theta(n^0 \lg n)$$

Hence: $T(n) = \Theta(\lg n)$

Master Method: Example 5

$$T(n) = 4\,T\left(\frac{n}{2}\right) + n^3$$

Master Method: Example 5

$$T(n) = 4T\left(\frac{n}{2}\right) + n^3$$

$a = 4$

$b = 2$

$f(n) = n^3$

$n^{\log_b a} \Rightarrow n^{\log_2 4} \Rightarrow n^2$

Master Method: Example 5

$$T(n) = 4T\left(\frac{n}{2}\right) + n^3$$

$a = 4$

$b = 2$

$f(n) = n^3$

$n^{\log_b a} \Rightarrow n^{\log_2 4} \Rightarrow n^2$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) > n^{\log_b a}$ so case 3 is applied. $\left[f(n) = \Omega(n^{\log_b a+\varepsilon})\right]$

Verify Regularity Condition:

$$a \cdot f\left(\frac{n}{b}\right) \le cf(n)$$

$$4 \cdot f\left(\frac{n}{2}\right) \le cn^3$$

$$4 \cdot \frac{n^3}{8} \le cn^2$$

$$\frac{1}{2} \le c$$

Master Method: Example 5

$$T(n) = 4\,T\left(\frac{n}{2}\right) + n^3$$

$a = 4$

$b = 2$

$f(n) = n^3$

$n^{\log_b a} \Longrightarrow n^{\log_2 4} \Longrightarrow n^2$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) > n^{\log_b a}$ so case 3 is applied. $\left[f(n) = \Omega(n^{\log_b a+\varepsilon})\right]$

$\Rightarrow T(n) = \Theta(f(n))$

$= \Theta(n^3)$

Hence: $T(n) = \Theta(n^3)$

Verify Regularity Condition:

$$a \cdot f\left(\frac{n}{b}\right) \le cf(n)$$

$$4 \cdot f\left(\frac{n}{2}\right) \le cn^3$$

$$4 \cdot \frac{n^3}{8} \le cn^2$$

$$\frac{1}{2} \le c$$

Master Method: Example 6

$$T(n) = T\left(\frac{n}{2}\right) + n^2$$

Master Method: Example 6

$$T(n) = T\left(\frac{n}{2}\right) + n^2$$

$a = 1$

$b = 2$

$f(n) = n^2$

$n^{\log_b a} \Rightarrow n^{\log_2 1} \Rightarrow n^0$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) > n^{\log_b a}$ so case 3 is applied. $\left[f(n) = \Omega(n^{\log_b a+\varepsilon})\right]$

$\Rightarrow T(n) = \Theta(f(n))$

$= \Theta(n^2)$

Hence: $T(n) = \Theta(n^2)$

Verify Regularity Condition:

$$a \cdot f\left(\frac{n}{b}\right) \leq cf(n)$$

$$1 \cdot f\left(\frac{n}{2}\right) \leq cn^2$$

$$\frac{n^2}{4} \leq cn^2$$

$$\frac{1}{4} \leq c$$

Master Method: Example 7

$$T(n) = 4T\left(\frac{n}{2}\right) + n^2$$

Master Method: Example 7

$$T(n) = 4T\left(\frac{n}{2}\right) + n^2$$

$a = 4$

$b = 2$

$f(n) = n^2$

$n^{\log_b a} \Longrightarrow n^{\log_2 4} \Longrightarrow n^2$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) = n^{\log_b a}$ so case 2 is applied. $\left[f(n) = \Theta(n^{\log_b a})\right]$

$$\Rightarrow T(n) = \Theta\left(n^{\log_b a} \lg n\right)$$

$$= \Theta\left(n^{\log_2 4} \lg n\right)$$

$$= \Theta(n^2 \lg n)$$

$Hence: T(n) = \Theta(n^2 \lg n)$

Master Method: Example 8

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

Master Method: Example 8

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

$a = 7$

$b = 3$

$f(n) = n^2$

$n^{\log_b a} \Rightarrow n^{\log_3 7} \Rightarrow n^{1.77}$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) > n^{\log_b a}$ so case 3 is applied. $\left[f(n) = \Omega(n^{\log_b a+\varepsilon})\right]$

$\Rightarrow T(n) = \Theta(f(n))$

Verify Regularity Condition:

$$a \cdot f\left(\frac{n}{b}\right) \le cf(n)$$

$$7 \cdot f\left(\frac{n}{3}\right) \le cn^2$$

$$7 \cdot \frac{n^2}{9} \le cn^2$$

$$\frac{7}{9} \le c$$

Master Method: Example 8

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

$a = 7$

$b = 3$

$f(n) = n^2$

$n^{\log_b a} \Rightarrow n^{\log_3 7} \Rightarrow n^{1.77}$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) > n^{\log_b a}$ so case 3 is applied. $[f(n) = \Omega(n^{\log_b a+\varepsilon})]$

$\Rightarrow T(n) = \Theta(f(n))$

$= \Theta(n^2)$

Hence: $T(n) = \Theta(n^2)$

Verify Regularity Condition:

$$a \cdot f\left(\frac{n}{b}\right) \le cf(n)$$

$$7 \cdot f\left(\frac{n}{3}\right) \le cn^2$$

$$7 \cdot \frac{n^2}{9} \le cn^2$$

$$\frac{7}{9} \le c$$

Master Method: Example 9

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

Master Method: Example 9

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

$a = 7$

$b = 2$

$f(n) = n^2$

$n^{\log_b a} \Longrightarrow n^{\log_2 7} \Longrightarrow n^{2.81}$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) < n^{\log_b a}$ so case 1 is applied. $\left[f(n) = O(n^{\log_b a - \varepsilon})\right]$

$\Rightarrow T(n) = \Theta\left(n^{\log_b a}\right)$

$= \Theta\left(n^{\log_2 7}\right)$

Hence: $T(n) = \Theta\left(n^{\log_2 7}\right)$

Master Method: Example 10

$$T(n) = 2\,T\left(\frac{n}{2}\right) + \sqrt{n}$$

Master Method: Example 10

$$T(n) = 2\,T\left(\frac{n}{2}\right) + \sqrt{n}$$

$a = 2$

$b = 2$

$f(n) = n^{1/2}$

$n^{\log_b a} \Rightarrow n^{\log_2 2} \Rightarrow n$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) < n^{\log_b a}$ so case 1 is applied. $\left[f(n) = O(n^{\log_b a-\varepsilon})\right]$

$$\Rightarrow T(n) = \Theta\left(n^{\log_b a}\right)$$

$$= \Theta\left(n^{\log_2 2}\right)$$

Hence: $T(n) = \Theta(n)$

Master Method: Example 11

$$T(n) = 3T\left(\frac{n}{4}\right) + n\lg n$$

Master Method: Example 11

$$T(n) = 3T\left(\frac{n}{4}\right) + n\lg n$$

$a = 3$

$b = 4$

$f(n) = n\lg n$

$n^{\log_b a} \Rightarrow n^{\log_4 3} \Rightarrow n^{0.79}$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) > n^{\log_b a}$ so case 3 is applied. $\left[f(n) = \Omega(n^{\log_b a+\varepsilon})\right]$

Verify Regularity Condition:

$$a \cdot f\left(\frac{n}{b}\right) \le cf(n)$$

$$3 \cdot f\left(\frac{n}{4}\right) \le cn\lg n$$

$$3 \cdot \frac{n}{4}\lg\frac{n}{4} \le cn\lg n$$

$$\frac{3}{4}[\lg n - 2] \le c\lg n$$

$$\frac{3}{4} \le c$$

Master Method: Example 11

$$T(n) = 3T\left(\frac{n}{4}\right) + n\lg n$$

$a = 3$

$b = 4$

$f(n) = n\lg n$

$n^{\log_b a} \Rightarrow n^{\log_4 3} \Rightarrow n^{0.79}$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) > n^{\log_b a}$ so case 3 is applied. $\left[f(n) = \Omega(n^{\log_b a+\varepsilon})\right]$

$\Rightarrow T(n) = \Theta(f(n))$

Verify Regularity Condition:

$$a \cdot f\left(\frac{n}{b}\right) \le cf(n)$$

$$3 \cdot f\left(\frac{n}{4}\right) \le cn\lg n$$

$$3 \cdot \frac{n}{4}\lg\frac{n}{4} \le cn\lg n$$

$$\frac{3}{4}[\lg n - 2] \le c\lg n$$

$$\frac{3}{4} \le c$$

Master Method: Example 11

$$T(n) = 3T\left(\frac{n}{4}\right) + n \lg n$$

$a = 3$

$b = 4$

$f(n) = n \lg n$

$n^{\log_b a} \Rightarrow n^{\log_4 3} \Rightarrow n^{0.79}$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) > n^{\log_b a}$ so case 3 is applied. $\left[f(n) = \Omega(n^{\log_b a+\varepsilon})\right]$

$\Rightarrow T(n) = \Theta(f(n))$

Verify Regularity Condition:

$$a \cdot f\left(\frac{n}{b}\right) \le cf(n)$$

$$3 \cdot f\left(\frac{n}{4}\right) \le cn \lg n$$

$$3 \cdot \frac{n}{4} \lg \frac{n}{4} \le cn \lg n$$

$$\frac{3}{4}[\lg n - 2] \le c \lg n$$

$$\frac{3}{4} \le c$$

Master Method: Example 11

$$T(n) = 3T\left(\frac{n}{4}\right) + n\lg n$$

$a = 3$

$b = 4$

$f(n) = n\lg n$

$n^{\log_b a} \Longrightarrow n^{\log_4 3} \Longrightarrow n^{0.79}$

Compare $f(n)$ and $n^{\log_b a}$:

$\Rightarrow f(n) > n^{\log_b a}$ so case 3 is applied. $\left[f(n) = \Omega(n^{\log_b a+\varepsilon})\right]$

$\Rightarrow T(n) = \Theta(f(n))$

$= \Theta(n\lg n)$

Hence: $T(n) = \Theta(n\lg n)$

Verify Regularity Condition:

$$a \cdot f\left(\frac{n}{b}\right) \le cf(n)$$

$$3 \cdot f\left(\frac{n}{4}\right) \le cn\lg n$$

$$3 \cdot \frac{n}{4}\lg\frac{n}{4} \le cn\lg n$$

$$\frac{3}{4}[\lg n - 2] \le c\lg n$$

$$\frac{3}{4} \le c$$

Master Method: Example 12

$$T(n) = 4T\left(\frac{n}{2}\right) + \frac{n^2}{\lg n}$$

$a = 4$

$b = 2$

$f(n) = n^2/\lg n$

$n^{\log_b a} \Longrightarrow n^{\log_2 4} \Longrightarrow n^2$

Compare $f(n)$ and $n^{\log_b a}$:

$\Longrightarrow$ Non-polynomial difference between $f(n)$ and $n^{\log_b a}$. Master method does not apply.

The difference must be polynomially larger by a factor of n^{ε} where $\varepsilon > 0$.

In this case the difference is only larger by a factor of $1/\lg n$.

Master Method: Example 13

$$T(n) = 2T\left(\frac{n}{2}\right) + n \lg n$$

Master Method: Example 13

$$T(n) = 2T\left(\frac{n}{2}\right) + n \lg n$$

$a = 2$

$b = 2$

$f(n) = n \lg n$

$n^{\log_b a} \Rightarrow n^{\log_2 2} \Rightarrow n$

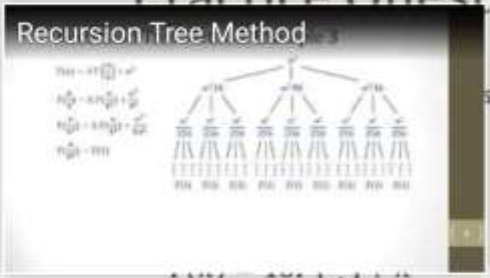
Compare $f(n)$ and $n^{\log_b a}$: Seems like case 3 should apply.

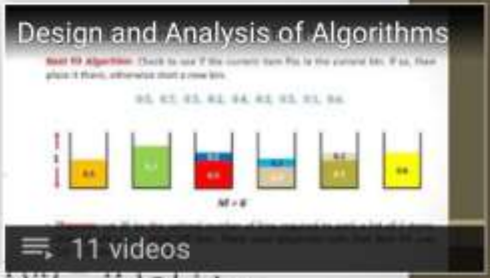
$\Rightarrow$ Master method does not apply. Non-polynomial difference between $f(n)$ and $n^{\log_b a}$.

The difference must be polynomially larger by a factor of n^{ε} where $\varepsilon > 0$.

In this case the difference is only larger by a factor of $\lg n$.

Practice Questions





s using t

$T(n) = 2T\left(\frac{n}{4}\right) + n^{0.51}$

$T(n) = T\left(\frac{2n}{5}\right) + n$ $\quad T(n) = 4T\left(\frac{n}{2}\right) + n$ $\quad T(n) = 0.5T\left(\frac{n}{2}\right) + \frac{1}{n}$

$T(n) = 3T\left(\frac{n}{2}\right) + n$ $\quad T(n) = 3T\left(\frac{n}{3}\right) + n$ $\quad T(n) = 4T\left(\frac{n}{2}\right) + n \lg n$

$T(n) = 3T\left(\frac{n}{3}\right) + 1$ $\quad T(n) = nT\left(\frac{n}{2}\right) + n$ $\quad T(n) = 6T\left(\frac{n}{3}\right) + n^2 \lg n$

$T(n) = 16T\left(\frac{n}{4}\right) + n$ $\quad T(n) = 2T\left(\frac{n}{2}\right) + 2^n$ $\quad T(n) = 64T\left(\frac{n}{8}\right) + n^2 \lg n$

The Master Method

Dr. Hasan Jamal

Practice Questions

Solve the following recurrences using the Master Method.

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

$T(n) = 16T\left(\frac{n}{4}\right) + n^2$

$T(n) = T\left(\frac{2n}{5}\right) + n$

$T(n) = 3T\left(\frac{n}{2}\right) + n$

$T(n) = 3T\left(\frac{n}{3}\right) + 1$

$T(n) = T\left(\frac{n}{2}\right) + n$

$T(n) = 2T\left(\frac{n}{4}\right) + n^{0.51}$

$T(n) = 4T\left(\frac{n}{2}\right) + n$

$T(n) = 3T\left(\frac{n}{3}\right) + n$

$T(n) = 4T\left(\frac{n}{2}\right) + \frac{n}{\lg n}$

$T(n) = 0.5T\left(\frac{n}{2}\right) + \frac{1}{n}$

$T(n) = 4T\left(\frac{n}{2}\right) + n \lg n$

$T(n) = 6T\left(\frac{n}{3}\right) + n^2 \lg n$

31:26 / 31:47

Screenshot has been saved to Pictures/ Screenshot