Lecture 4

RECURRENCE TREE METHOD

RECAP

- Derive Recurrence Relation from the algorithm
- Solve the Recurrence relation using Substitution Method

RECURRENCE TREE METHOD

- A recursion tree models the costs (time) of a recursive execution of an algorithm.
- The recursion tree method is good for generating guesses for the substitution method.
- The recursion-tree method promotes intuition.

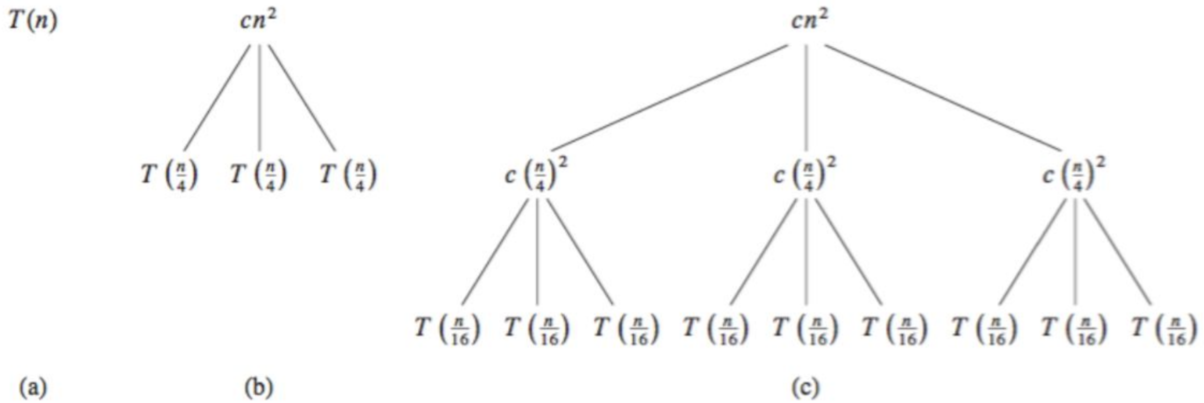
Convert the recurrence into a tree:

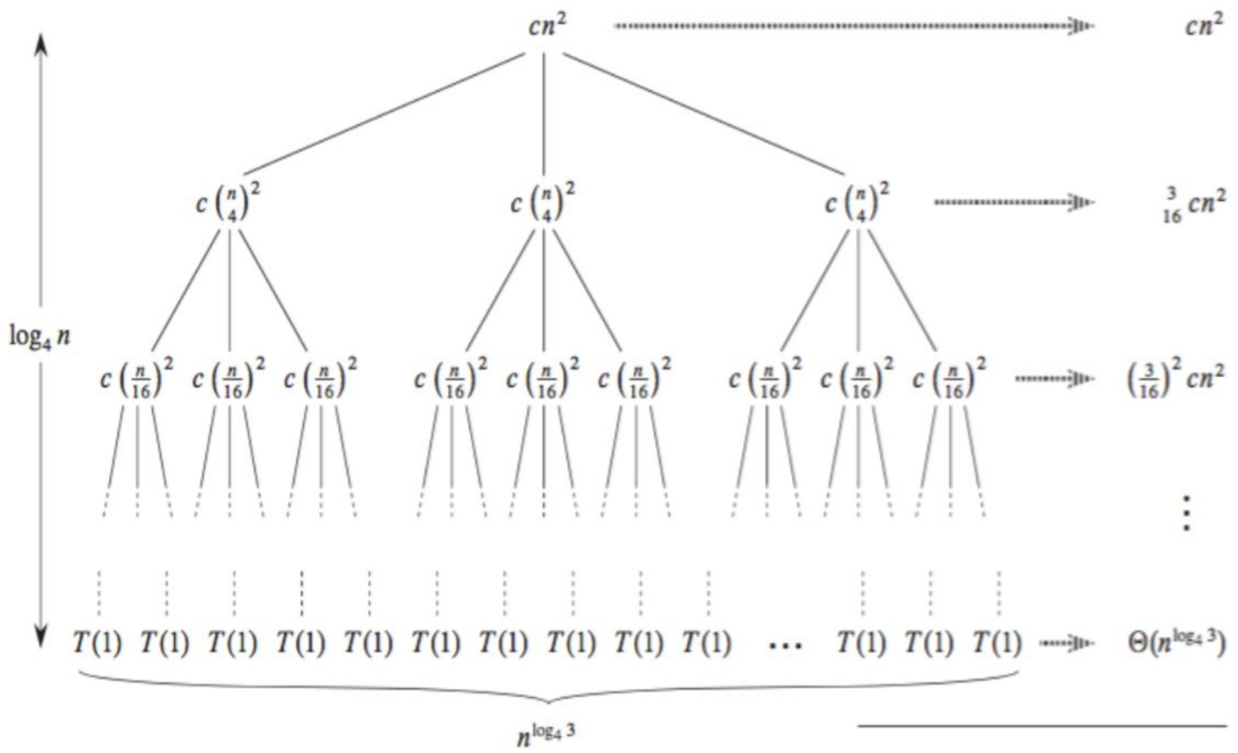
- Each node represents the cost incurred at various levels of recursion
- Sum up the costs of all levels

Solve the Recurrence Relation $T(n)=3T(\lfloor n/4 \rfloor)+\Theta(n^2)$ using Recurrence tree method.

<u>Solution:</u>

We drop the floors and write a recursion tree for $T(n) = 3T(n/4) + cn^2$




cn^2
$\log_4 n$
$c\left(\frac{n}{4}\right)^2$
$c\left(\frac{n}{16}\right)^2$
$T(1)$
$n^{\log_4 3}$
$\frac{3}{16}cn^2$
$\left(\frac{3}{16}\right)^2 cn^2$
$\Theta(n^{\log_4 3})$

(d)

Total: $O(n^2)$

Total Time Complexity:

$$
\begin{aligned}
T(n) &= cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \cdots + \left(\frac{3}{16}\right)^{\log_4 n - 1} cn^2 + \Theta(n^{\log_4 3}) \\
&= \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3})
\end{aligned}
$$

The left term is just the sum of a geometric series. So $T(n)$ evaluates to

$$\frac{(3/16)^{\log_4 n} - 1}{(3/16) - 1} cn^2 + \Theta(n^{\log_4 3})$$

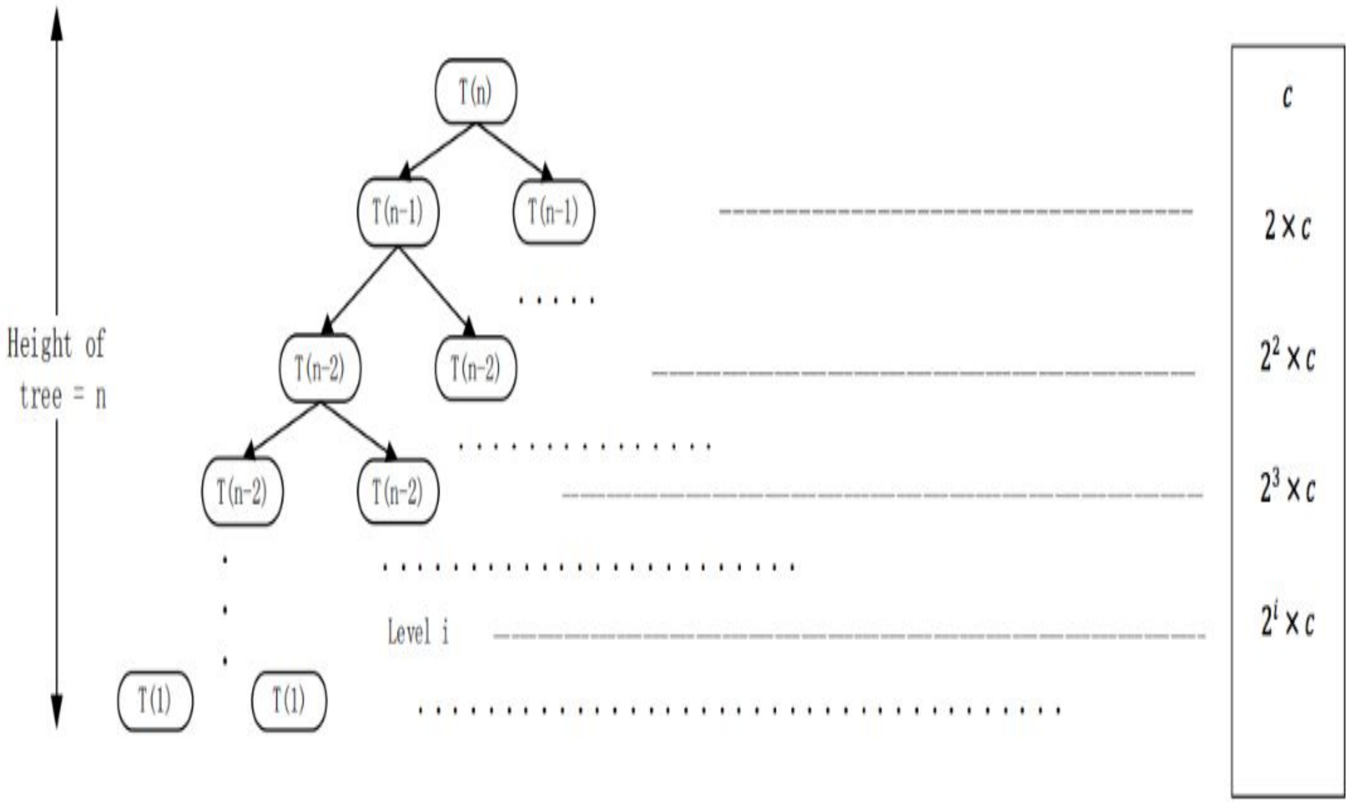
This looks complicated but we can bound it (from above) by the sum of the infinite series

$$\sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) = \frac{1}{1 - (3/16)} cn^2 + \Theta(n^{\log_4 3})$$

Since functions in $\Theta(n^{\log_4 3})$ are also in $O(n^2)$, this whole expression is $O(n^2)$. Therefore, we can guess that $T(n) = O(n^2)$.

Ex:2

$T(n)=\begin{cases} 2T(n-1)+c, \\ T(1)=1 \end{cases}$



$$T(n) = \sum_{i=0}^{n-1} (2)^i c = c\,\frac{2^n - 1}{2 - 1} = c(2^n - 1)$$

$$T(n) = \Theta(2^n - 1) = \Theta(2^n - 1) = \Theta(2^n)$$