Recursion Trees

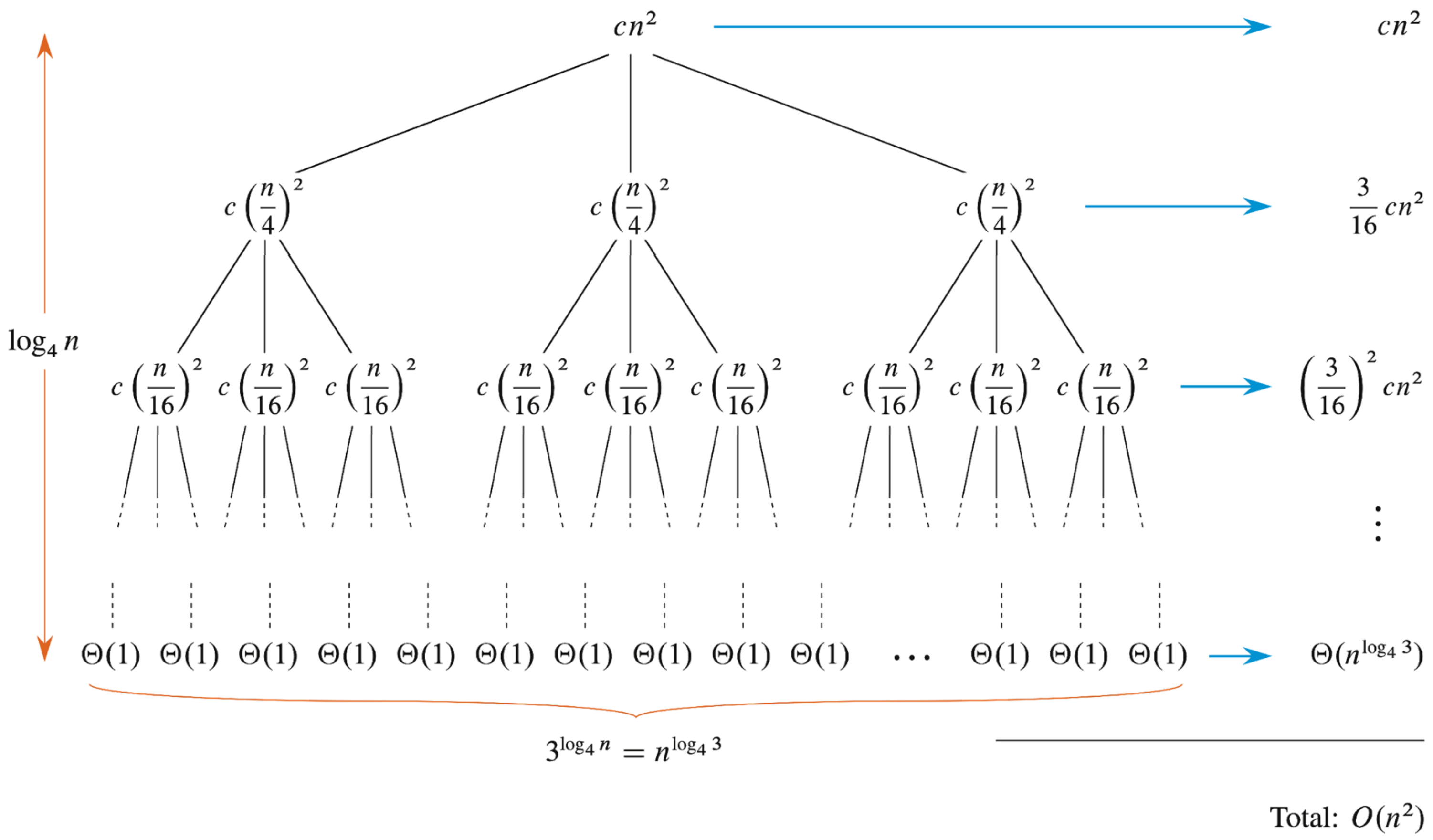
Adapted from the CLRS book slides

RECURSION TREES

Use to generate a guess. Then verify by substitution method.

Example: $T(n) = 3T(n/4) + \Theta(n^2)$.

Draw out a recursion tree for $T(n) = 3T(n/4) + cn^2$:



For simplicity, assume that n is a power of 4 and the base case is $T(1) = \Theta(1)$.

RECURSION TREES (continued)

Subproblem size for nodes at depth i is $n/4^i$. Get to base case when $n/4^i = 1 \Rightarrow n = 4^i \Rightarrow i = \log_4 n$.

Each level has 3 times as many nodes as the level above, so that depth i has 3^i nodes. Each internal node at depth i has cost $c(n/4^i)^2 \Rightarrow$ total cost at depth i (except for leaves) is $3^i c(n/4^i)^2 = (3/16)^i cn^2$. Bottom level has depth $\log_4 n \Rightarrow$ number of leaves is $3^{\log_4 n} = n^{\log_4 3}$. Since each leaf contributes $\Theta(1)$, total cost of leaves is $\Theta(n^{\log_4 3})$.

Add up costs over all levels to determine cost for the entire tree:

$$
\begin{aligned}
T(n) &= \sum_{i=0}^{\log_4 n} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\
&< \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\
&= \frac{1}{1-(3/16)} cn^2 + \Theta(n^{\log_4 3}) \\
&= \frac{16}{13} cn^2 + \Theta(n^{\log_4 3}) \\
&= O(n^2)\,.
\end{aligned}
$$

Idea: Coefficients of cn^2 form a decreasing geometric series. Bound it by an infinite series, and get a bound of 16/13 on the coefficients.

RECURSION TREES (continued)

Use substitution method to verify $O(n^2)$ upper bound. Show that $T(n) \le dn^2$ for constant $d > 0$:

$$\begin{aligned} T(n) &\le 3T(n/4) + cn^2 \\ &\le 3d(n/4)^2 + cn^2 \\ &= \frac{3}{16}dn^2 + cn^2 \\ &\le dn^2 , \end{aligned}$$

by choosing $d \ge (16/13)c$. *[Again, we get to name but not choose c, and we get to name and choose d.]*

That gives an upper bound of $O(n^2)$. The lower bound of $\Omega(n^2)$ is obvious because the recurrence contains a $\Theta(n^2)$ term. Hence, $T(n) = \Theta(n^2)$.

RECURSION TREES (continued)

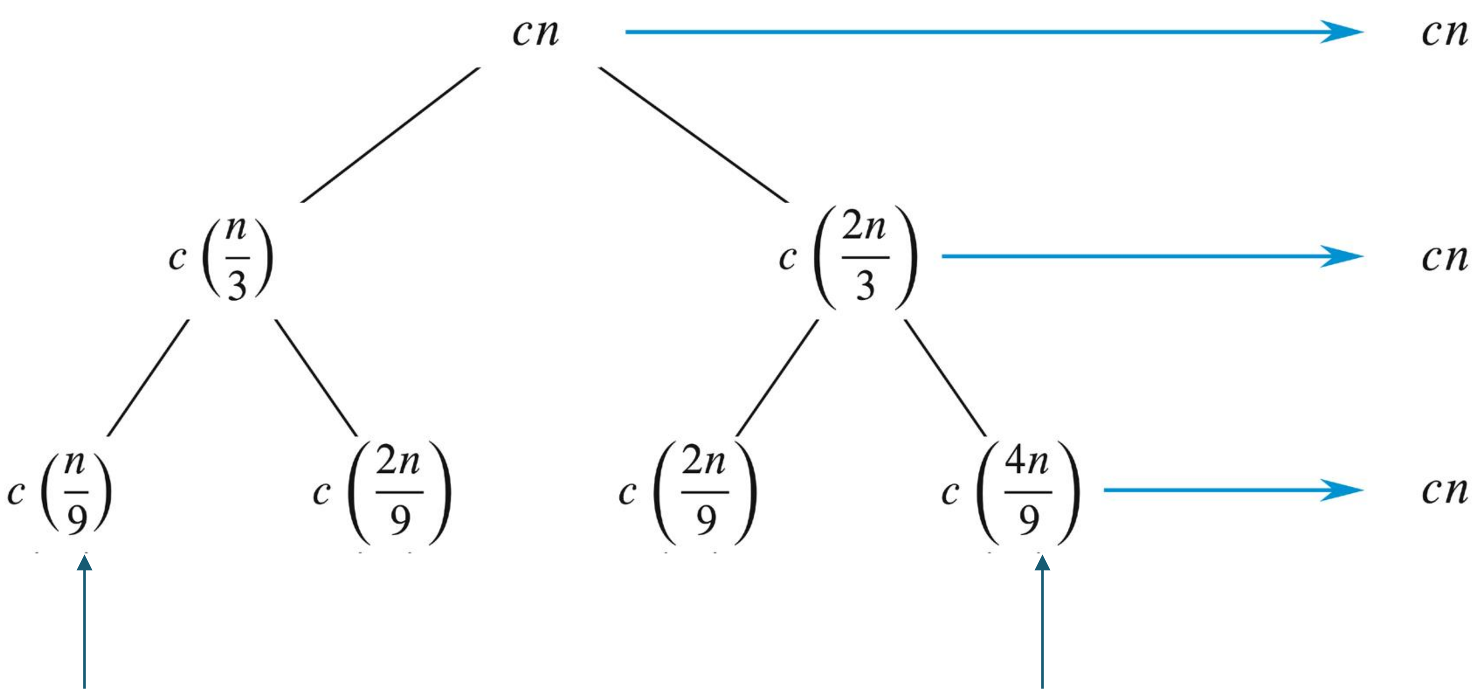
Irregular Example:

$T(n) = T(n/3) + T(2n/3) + \Theta(n)$.
For upper bound, rewrite as $T(n) \le T(n/3) + T(2n/3) + cn$; for lower bound, as $T(n) \ge T(n/3) + T(2n/3) + cn$.

By summing across each level, the recursion tree shows the cost at each level of recursion (minus the costs of recursive calls, which appear in subtrees):

RECURSION TREES (continued)

Irregular Example:



leftmost branch reaches $n = 1$ after $\log_3 n$ levels

rightmost branch reaches $n = 1$ after $\log_{\frac{3}{2}} n$ levels

RECURSION TREES (continued)

Irregular Example:

- There are $\log_3 n$ full levels (going down the left side), and after $\log_{3/2} n$ levels, the problem size is down to 1 (going down the right side).
- Each full level contributes at least/most cn.
- Lower bound guess: $\geq d\, n \log_3 n = \Omega(n \lg n)$ for some positive constant d.
- Upper bound guess: $\leq d\, n \log_{3/2} n = O(n \lg n)$ for some positive constant d.
- Then ***prove*** by substitution.