Solving Recursion

Substitution method, recursion tree, master method

Recurrences

- A recurrence is an equation or inequality that describes a function in terms of its value on smaller inputs
- Recurrences not necessarily divide the subproblems into equal parts
- In the worst case, e.g. linear search, the recurrence might just reduce the problem size by 1
- We mostly solve recurrences to get θ bound or O bound

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

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

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

$T(n) \leq 2T\left(\frac{n}{2}\right) + \theta(n)$ $\Rightarrow$ only upper bound is possible $\Rightarrow$ $O-$ $notation$

$T(n) \geq 2T\left(\frac{n}{2}\right) + \theta(n) \Rightarrow$ only lower bound is possible $\Rightarrow$ $\Omega -$ $notation$

Solving Recurrences

- **Substitution Method**: guess a bound then use mathematical induction to prove that the guess is correct
- **Recursion-tree Method**: convert the recurrence into a tree whose nodes represent the costs incurred at various levels of the recursion. Then, use techniques to bound the summations
- **Master Method**: $T(n) = aT\left(\frac{n}{b}\right) + f(n)$, three conditions need to be remembered to get solutions for the recurrences that take above type of form

Recursion Tree Method

- Serves as a straight forward to making a good guess for many problems
- Each node represents the cost of a single sub-problem in the set of recursive function invocations
- Cost of each sub-problem is summed to get cost-per level
- Cost of each level is summed to get the total cost of recursion
- While using recursion tree to generate a good guess minor variations in the boundary conditions are tolerable

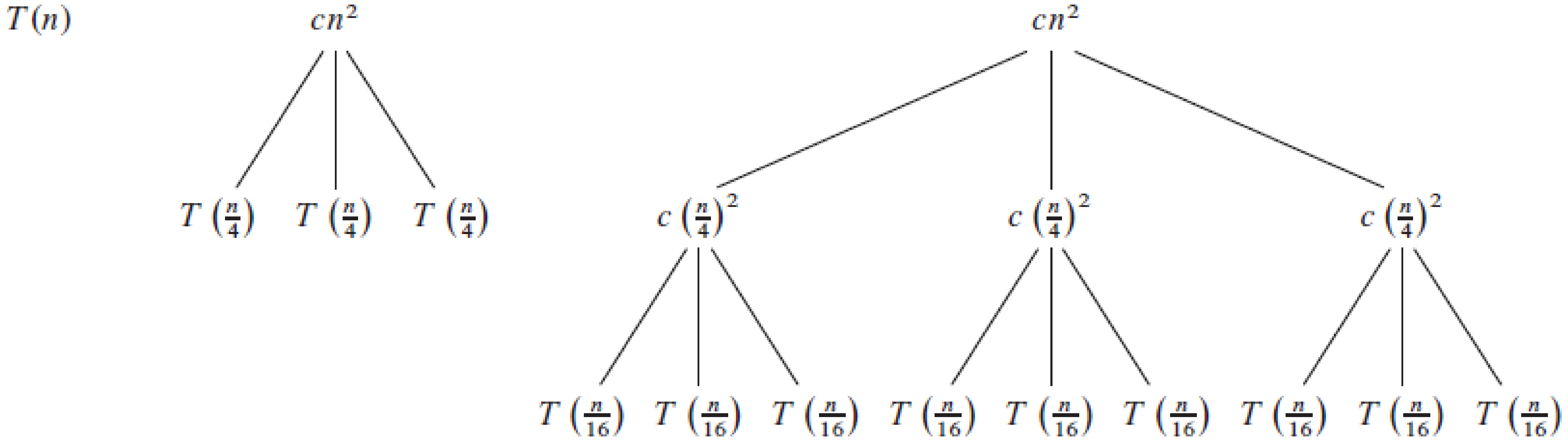
Example

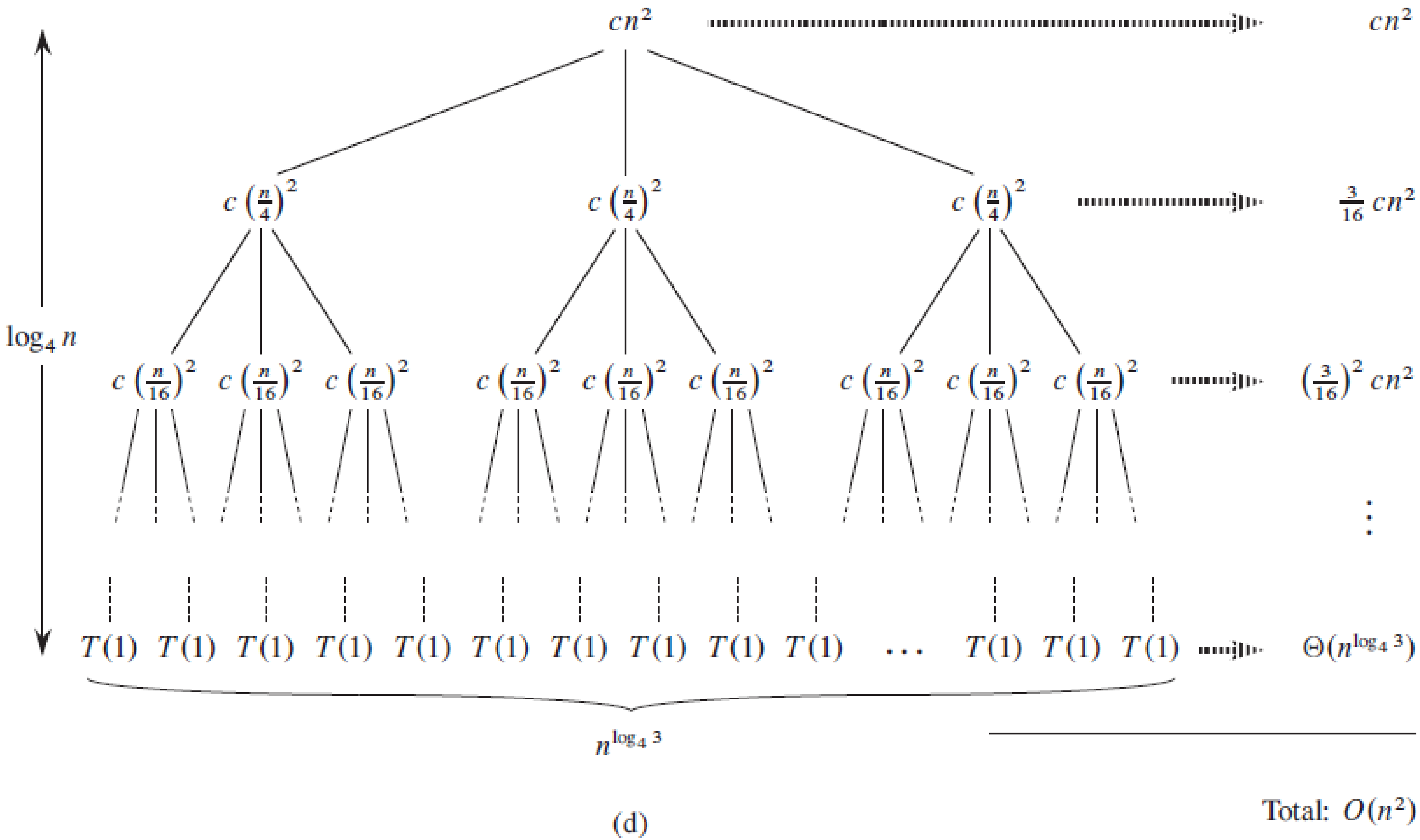
$$T(n) = 3T(\lfloor n/4 \rfloor) + \theta(n^2)$$

- Indicates division of problem into three smaller problems at each step
- Each sub-problem is 1/4th the size of original problem
- Merger step takes n2 time at each step

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

- The recursion tree for this problem can be designed as follows


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



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

(d)

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

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

Verification using Substitution Method

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

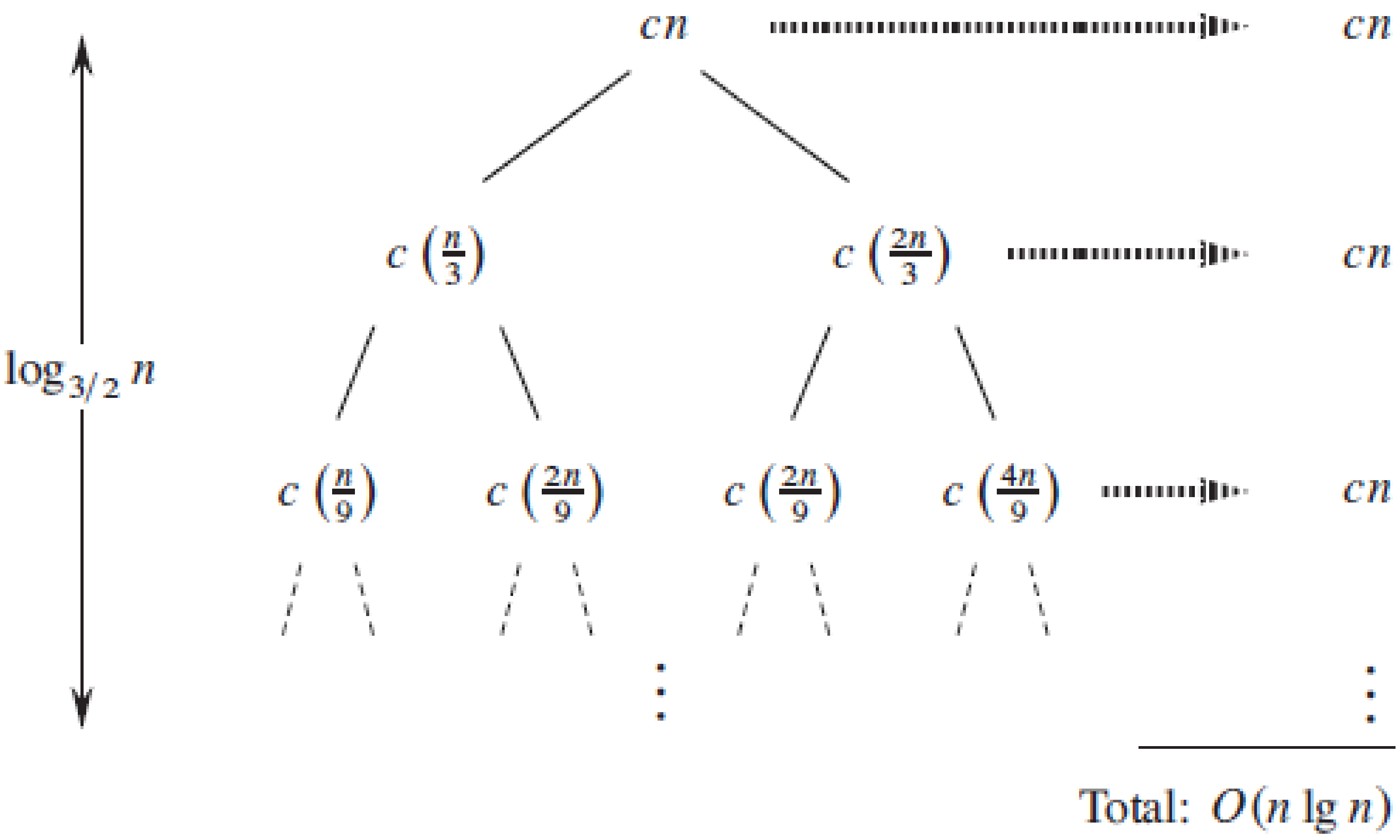
where the last step holds as long as $d \ge (16/13)c$.

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

- Each time the problem is divided asymmetrically into sub-problems of sizes n/3 and 2n/3
- Merger steps takes O(n) time
- The equation can be re-written as:

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

- We use this to draw the recursion tree



cn
cn
$c\left(\frac{n}{3}\right)$
$c\left(\frac{2n}{3}\right)$
cn
$\log_{3/2} n$
$c\left(\frac{n}{9}\right)$
$c\left(\frac{2n}{9}\right)$
$c\left(\frac{2n}{9}\right)$
$c\left(\frac{4n}{9}\right)$
cn
Total: $O(n \lg n)$

- Height of the tree: longest sequence from root

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

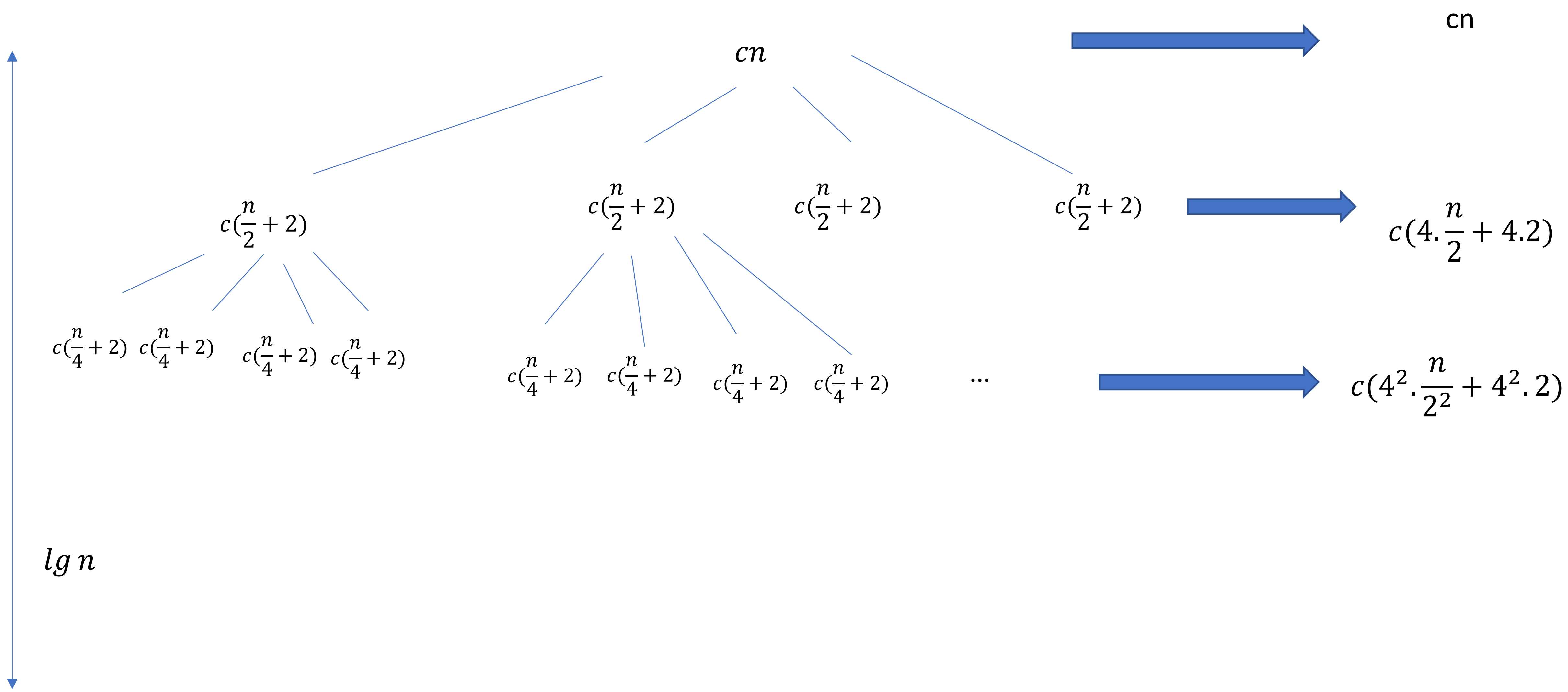
- Root contributes cn to cost
- If each level contributes equally, total cost is bounded by $O\left(cn \log_{\frac{3}{2}} n\right) \Rightarrow$ $\mathrm{O(nlgn)}$
- Each level however does not contribute equally
- Contribution of leaves is different
- No. of leaves = $2^{\log_{3/2} n} \Rightarrow \ n^{\log_{3/2} 2}$
- If each leaf contributes constant time then the bound is $\theta(n^{\log_{3/2} 2})$ for leaves
- $\log_{3/2} 2$ is strictly greater than 1, therefore bound is $\omega(nlgn)$

Verification using Substitution Method

$$
\begin{aligned}
T(n) &\le T(n/3) + T(2n/3) + cn \\
&\le d(n/3)\lg(n/3) + d(2n/3)\lg(2n/3) + cn \\
&= (d(n/3)\lg n - d(n/3)\lg 3) \\
&\qquad + (d(2n/3)\lg n - d(2n/3)\lg(3/2)) + cn \\
&= dn\lg n - d((n/3)\lg 3 + (2n/3)\lg(3/2)) + cn \\
&= dn\lg n - d((n/3)\lg 3 + (2n/3)\lg 3 - (2n/3)\lg 2) + cn \\
&= dn\lg n - dn(\lg 3 - 2/3) + cn \\
&\le dn\lg n ,
\end{aligned}
$$

Exercise

- Use a recursion tree to determine a good asymptotic upper bound on the recurrence
$$T(n) = 4T\left(\frac{n}{2} + 2\right) + n$$
- Use the substitution method to verify your answer.



cn
cn
$c(\frac{n}{2}+2)$ $c(\frac{n}{2}+2)$ $c(\frac{n}{2}+2)$ $c(\frac{n}{2}+2)$
$c(4.\frac{n}{2}+4.2)$
$c(\frac{n}{4}+2)$ $c(\frac{n}{4}+2)$ $c(\frac{n}{4}+2)$ $c(\frac{n}{4}+2)$
$c(\frac{n}{4}+2)$ $c(\frac{n}{4}+2)$ $c(\frac{n}{4}+2)$ $c(\frac{n}{4}+2)$ …
$c(4^2.\frac{n}{2^2}+4^2.2)$
$lg\, n$

Analysis

- Height of tree $\Rightarrow \frac{n}{2^k} = 1 \Rightarrow k = \log_2 n$
- Total cost at ith level $\Rightarrow c(4^i \times \frac{n}{2^i} + 4^i \times 2)$
- Cost of last level $\Rightarrow \; 4^k \times T(1) \Rightarrow \; 4^{\log_2 n} \Rightarrow \theta(n^2)$

Calculating total cost

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

$$T(n) = \sum_{i=0}^{lgn-1} c\left(4^i \times \frac{n}{2^i} + 4^i \times 2\right) - 2 + \theta(n^2)$$

$$= cn\left(2^{lgn-1} - 1\right) + \frac{2}{3}c\left(4^{lgn-1}\right) - 2 + \theta(n^2)$$

$$= cn\left(\frac{n}{2} - 1\right) + \frac{2}{3}c\left(\frac{n^2}{4}\right) + \theta(n^2) - 2$$

$$= cn^2 - \mathrm{cn} + \frac{1}{6}\mathrm{c}(n^2) + \theta(n^2) - 2$$

$$\Rightarrow O(n^2)$$

Verification using Substitution Method

- For this and all such problems, we will use the method of subtracting a lower order term from the equation for proving the bound
- Suppose, we guess the solution to be $O(n^2)$
- Then, we have to prove that $T(n) \leq c.n^2 - 6\mathrm{n}$
- Let us assume that the bound holds for m = n/2+2
- Then,

$$T\left(\frac{n}{2} + 2\right) \leq c\left(\frac{n^2}{4} + 4 + 2\mathrm{n}\right) - 6 \times \left(\frac{n}{2} + 2\right)$$

$$\Rightarrow T(n) \leq 4.c\left(\frac{n^2}{4} + 4 + 2\mathrm{n} - 3n - 12\right) + n$$

$$\Rightarrow T(n) \leq 4.\left[c\left(\frac{n^2}{4} + 4 + 2\mathrm{n}\right) - 6 \times \left(\frac{n}{2} + 2\right)\right] + n$$

$$= cn^2 + 16c + 8cn - 12n - 48 + n$$

$$= cn^2 + 16c + 8cn - 11n - 48$$

$$\Rightarrow 16c + 8cn - 11n \leq 0 \Rightarrow c = \frac{1}{8}$$

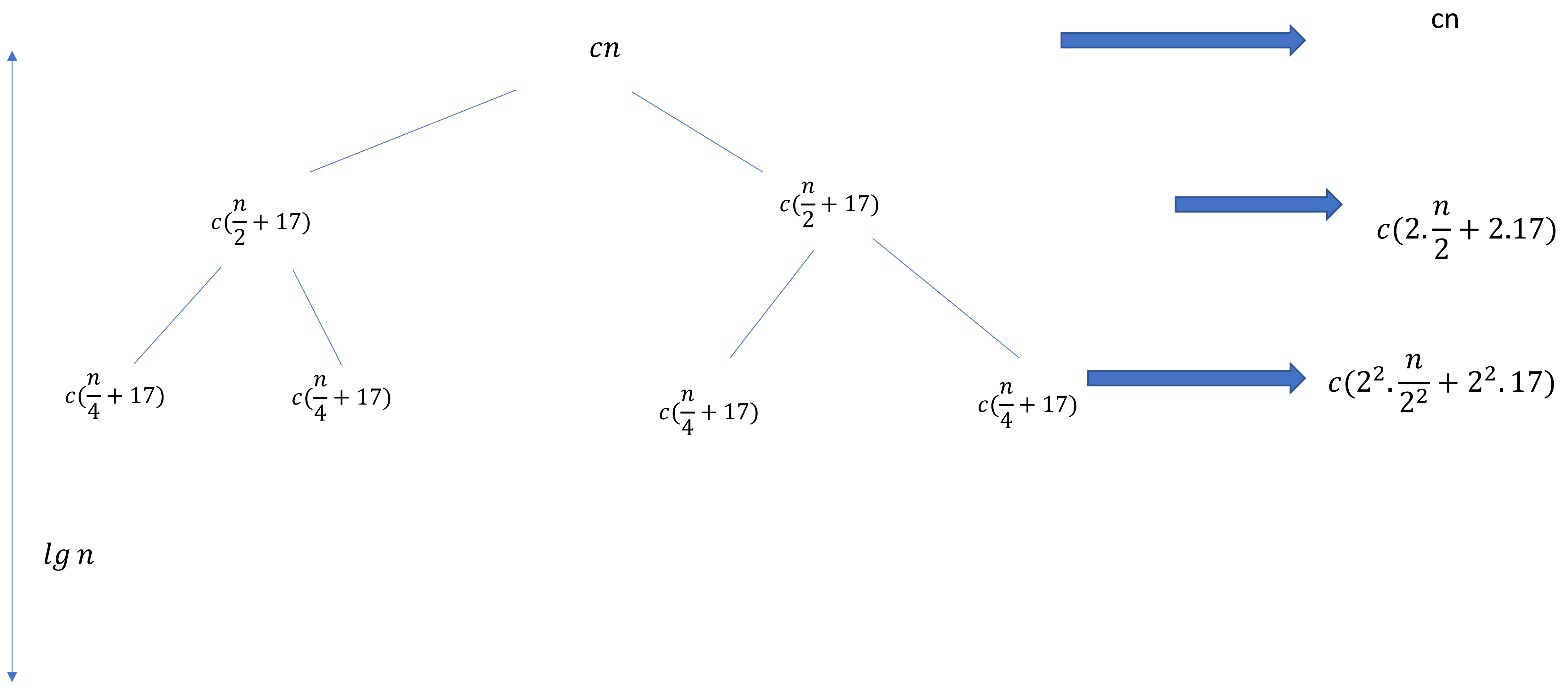
Therefore,

$$T(n) = O(n^2)$$

Example 2

$$T(n) = 2T\left(\frac{n}{2} + 17\right) + n$$

- Height of the tree $\Rightarrow k = \log_2 n$
- Total Cost at the ith level $\Rightarrow 2^i . c\left(\frac{n}{2^i} + 17\right)$
- Cost at the last level $\Rightarrow 2^i \times T(1) = \theta(n)$


cn
$c(\frac{n}{2} + 17)$
$c(\frac{n}{2} + 17)$
$c(\frac{n}{4} + 17)$
$c(\frac{n}{4} + 17)$
$c(\frac{n}{4} + 17)$
$c(\frac{n}{4} + 17)$
$lg\ n$
cn
$c(2.\frac{n}{2} + 2.17)$
$c(2^2.\frac{n}{2^2} + 2^2.17)$

$$T(n) = \sum_{i=0}^{\log_2 n-1} \left(cn + 2^i \times 17\right) - 17 + \theta(n)$$

$$T(n) = \frac{c}{2} nlogn + 17\left(\frac{n}{2} - 1\right) - 17 + \theta(n)$$

$$T(n) \leq cnlgn$$

Exercise

- $T(n) = 3T(\lfloor n/2 \rfloor) + n$

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