

CSE 202—Day 3

Recurrence relations (version 2)

Neil Rhodes

UC San Diego

Recurrence Relations

Given a recurrence relation, determine the asymptotic complexity. Three basic approaches:

Tree Method: Draw out the tree and count

Substitution Method: Guess complexity and prove by induction

Master Method For most recurrences of the form
$$T(n) = aT\left(\frac{n}{b}\right) + f(n),$$
 use formula

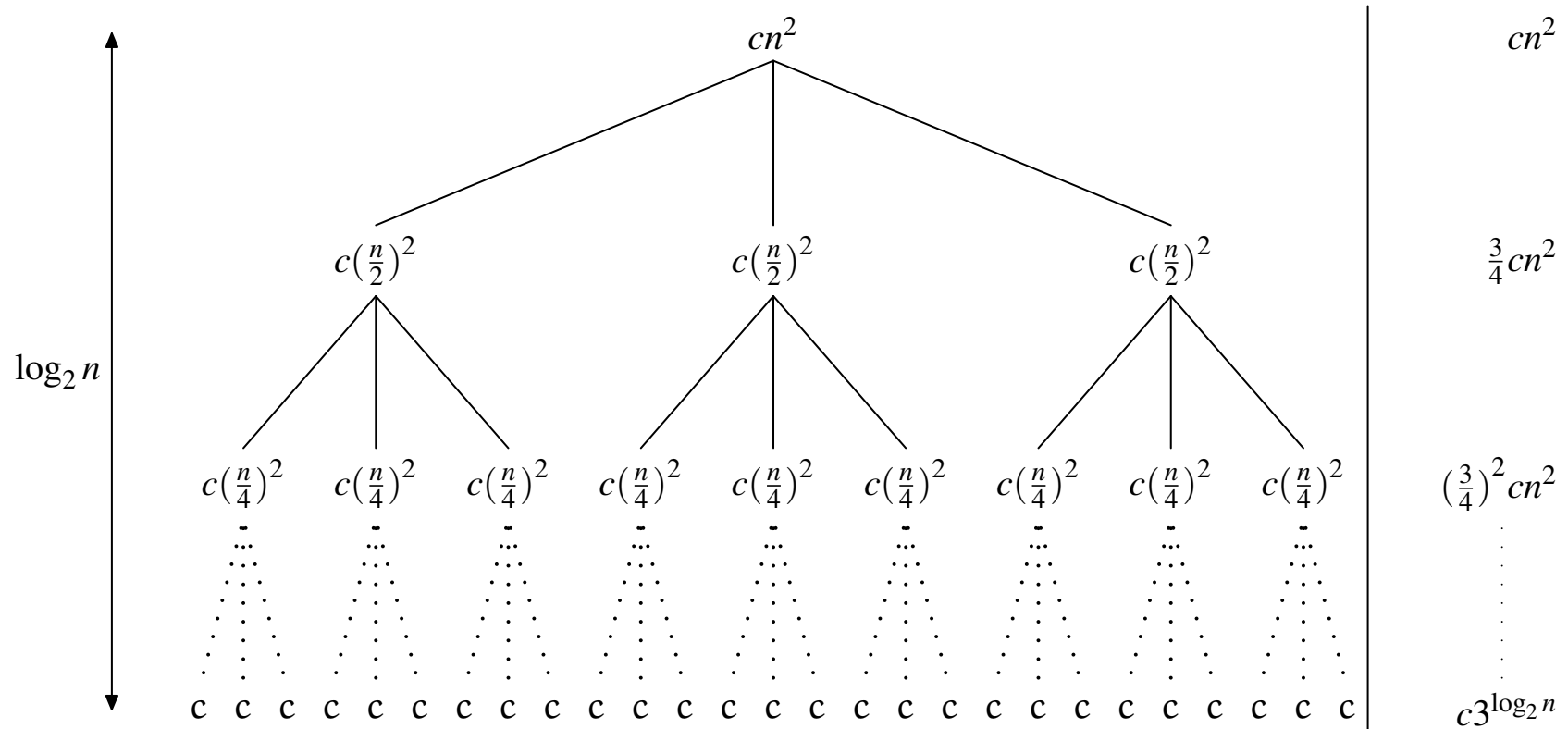
Recurrence Relations

Notes:

- For small n , $T(n) = \Theta(1)$. We leave off the base cases for most of the recurrences we see. Changing the constant changes the solution, but not the order of growth.
- T is defined only on natural numbers, so we must use $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ as necessary. Later, we'll relax their use.

Tree Method

$$T(n) = 3T(\lfloor \frac{n}{2} \rfloor) + O(n^2)$$



Tree Method

$$T(n) = 3T(\lfloor \frac{n}{2} \rfloor) + O(n^2)$$

$$\begin{aligned} T(n) &= cn^2 + cn^2 \left(\frac{3}{4}\right)^1 + cn^2 \left(\frac{3}{4}\right)^2 + \dots + c3^{\log_2 n} \\ &= cn^2 \left(\frac{3}{4}\right)^0 + cn^2 \left(\frac{3}{4}\right)^1 + \dots + c3^{\log_2 n} \left(\frac{n}{2^{\log_2 n}}\right)^2 \\ &= cn^2 \left(\frac{3}{4}\right)^0 + cn^2 \left(\frac{3}{4}\right)^1 + \dots + cn^2 \frac{3^{\log_2 n}}{4^{\log_2 n}} \\ &= \sum_{i=0}^{\log_2 n} cn^2 \left(\frac{3}{4}\right)^i = cn^2 \sum_{i=0}^{\log_2 n} \left(\frac{3}{4}\right)^i = cn^2 \Theta(1) \end{aligned}$$

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

Substitution Method

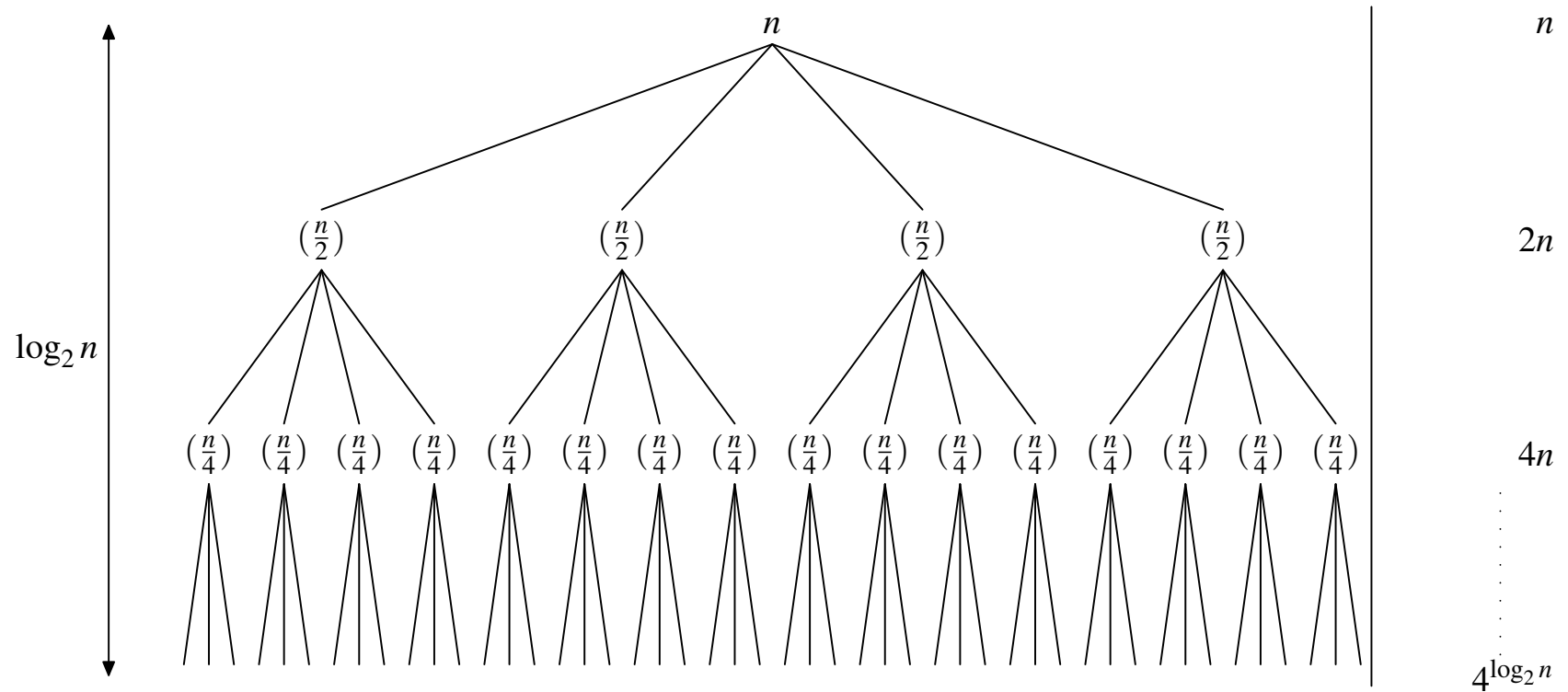
$T(n) = 3T(\lfloor \frac{n}{2} \rfloor) + n^2$ Assume $T(\lfloor \frac{n}{2} \rfloor) \leq c\lfloor \frac{n}{2} \rfloor^2$. Show $T(n) \leq cn^2$.

$$\begin{aligned} T(n) &= 3T(\lfloor \frac{n}{2} \rfloor) + n^2 \leq 3c\lfloor \frac{n}{2} \rfloor^2 + n^2 \\ &\leq 3c\frac{n^2}{4} + n^2 = n^2(\frac{3c}{4} + 1) \\ &\leq cn^2 \quad (\text{if } c \geq 4) \end{aligned}$$

Base case: $T(1) = k \leq cn^2$ (if $c \geq k$). So, $c = \max(4, k)$.

Tree Method

$$T(n) = 4T(\lfloor \frac{n}{2} \rfloor) + n$$



Tree Method

$$T(n) = 4T(\lfloor \frac{n}{2} \rfloor) + n$$

$$T(n) = n + 2n + 4n + \dots + 4^{\log_2 n}$$

$$= \sum_{i=0}^{\log_2 n} n2^i = n \sum_{i=0}^{\log_2 n} 2^i$$

$$= n2^{\log_2 n + 1}$$

$$= n(n + 1)$$

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

Substitution Method

$T(n) = 4T(\lfloor \frac{n}{2} \rfloor) + n$. Assume $T(\lfloor \frac{n}{2} \rfloor) \leq c\lfloor \frac{n}{2} \rfloor^2$. Show $T(n) \leq cn^2$.

$$\begin{aligned} T(n) &= 4T(\lfloor \frac{n}{2} \rfloor) + n \leq 4c\lfloor \frac{n}{2} \rfloor^2 + n \\ &\leq 4c\frac{n^2}{4} + n = cn^2 + n \\ &> cn^2 \end{aligned}$$

Uh-oh, inductive assumption not strong enough

Substitution Method

$T(n) = 4T(\lfloor \frac{n}{2} \rfloor) + n$. Solution: tighten the inductive assumption. Assume $T(\lfloor \frac{n}{2} \rfloor) \leq c\lfloor \frac{n}{2} \rfloor^2 - d\lfloor \frac{n}{2} \rfloor$. Show $T(n) \leq cn^2 - dn$.

$$\begin{aligned} T(n) &= 4T(\lfloor \frac{n}{2} \rfloor) + n \leq 4(c\lfloor \frac{n}{2} \rfloor^2 - d\lfloor \frac{n}{2} \rfloor) + n \\ &\leq 4c\frac{n^2}{4} - 4(d(\frac{n}{2} - 1)) + n = cn^2 - 2dn + 4d + n \\ &\leq cn^2 - dn \quad (\text{if } dn \geq n + 4d \rightarrow d \geq 2, n \geq 8, c > 0) \end{aligned}$$

Master Method Idea

A recurrence tree will, in general, have either

1. Time dominated by cost of the leaves

$$(T(n) = 4T(\lfloor \frac{n}{2} \rfloor) + n)$$

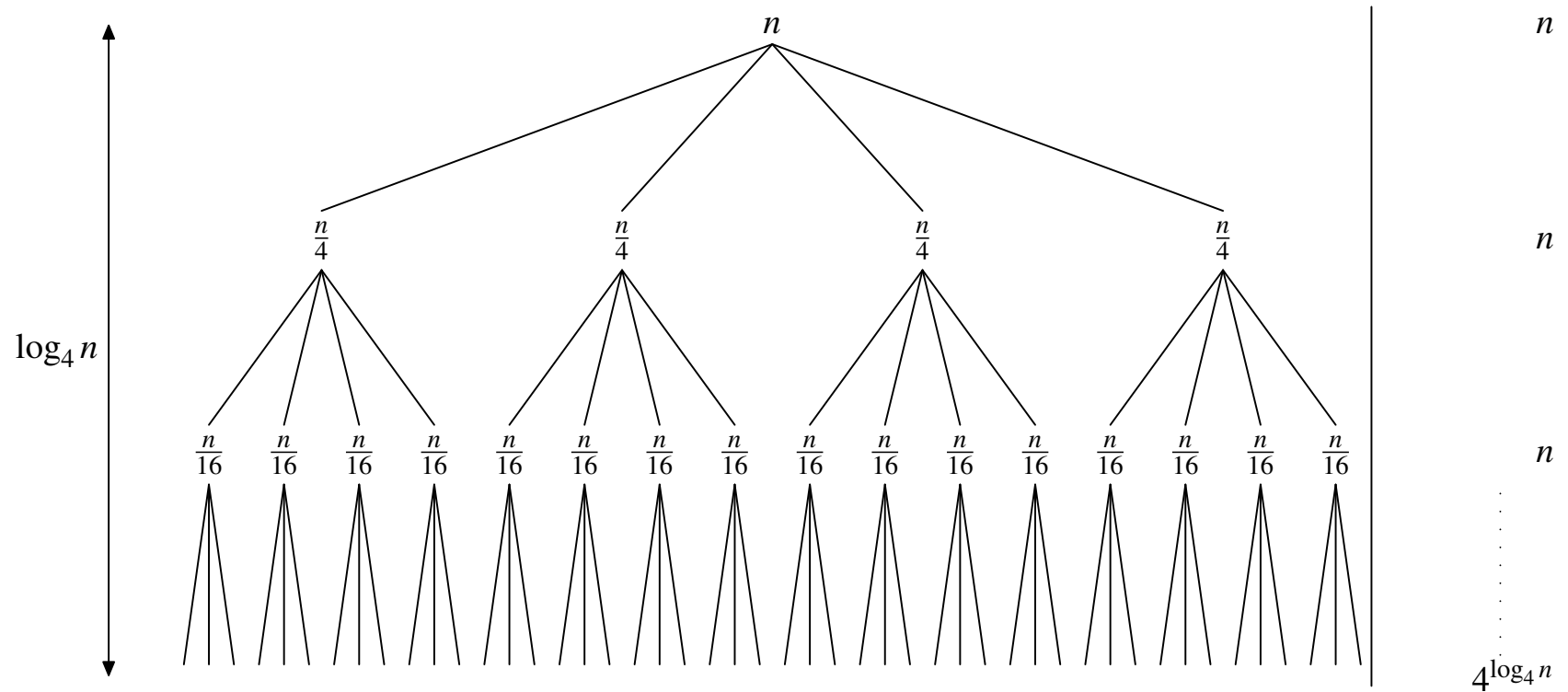
2. Time evenly distributed across the levels of the tree (next example)

3. Time dominated by the cost of the root

$$(T(n) = 3T(\lfloor \frac{n}{2} \rfloor) + O(n^2))$$

Tree Method

$$T(n) = 4T(\lfloor \frac{n}{4} \rfloor) + n$$



Tree Method

$$T(n) = 4T(\lfloor \frac{n}{4} \rfloor) + n$$

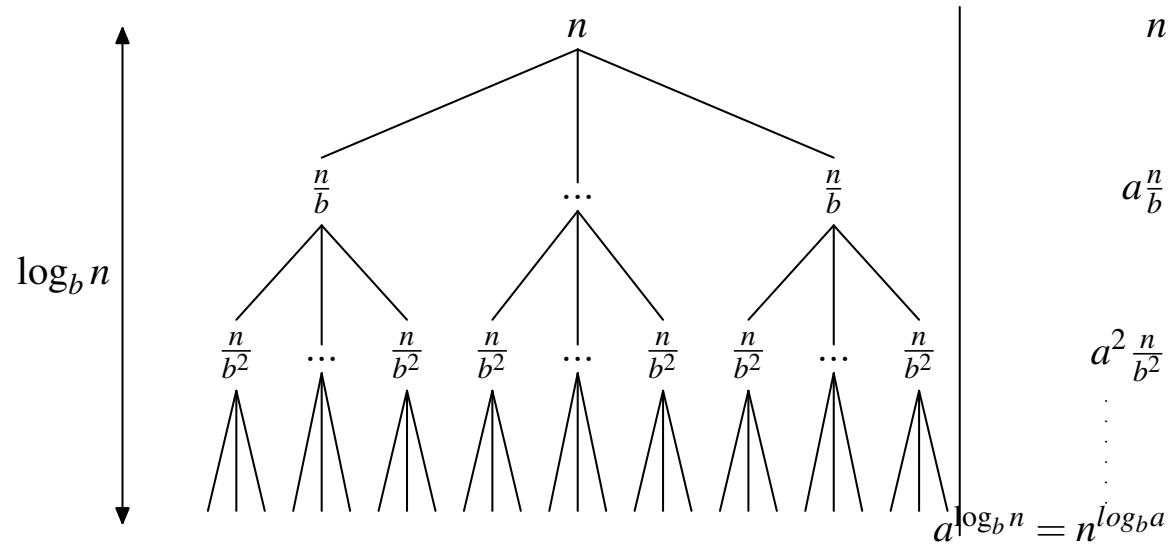
$$T(n) = n + n + n + \dots + 4^{\log_4 n}$$

$$= \sum_{i=0}^{\log_2 n} n = n \sum_{i=0}^{\log_2 n} 1 = n\Theta(\log n)$$

$$T(n) = \Theta(n \log n)$$

Limited Master Method

Recurrence: $T(n) = aT(n/b) + n$



$$\begin{aligned} T(n) &= n + \frac{a}{b}n + \frac{a^2}{b}n + \dots + a^{\log_b n}n \\ &= \sum_{i=0}^{\log_b n} \left(\frac{a}{b}\right)^i n = n \sum_{i=0}^{\log_b n} \left(\frac{a}{b}\right)^i \end{aligned}$$

Limited Master Method

Evaluate: $T(n) = n \sum_{i=0}^{\log_b n} \left(\frac{a}{b}\right)^i$

1. $\frac{a}{b} > 1$: Time dominated by leaves. $T(n) = n \sum_{i=0}^{\log_b n} \left(\frac{a}{b}\right)^i = n \frac{\left(\frac{a}{b}\right)^{\log_b(n+1)} - 1}{\frac{a}{b} - 1} = O\left(n\left(\frac{a}{b}\right)^{\log_b n}\right) = O\left(n \frac{a^{\log_b n}}{b^{\log_b n}}\right) = O(a^{\log_b n}) = O(n^{\log_b a})$
2. $\frac{a}{b} = 1$: Time evenly distributed across all levels. $T(n) = n \log n = O(n \log n)$.
3. $\frac{a}{b} < 1$: Time dominated by root. $n \sum_{i=0}^{\log_b n} \left(\frac{a}{b}\right)^i < n \frac{1}{1 - \frac{a}{b}} = O(n)$.

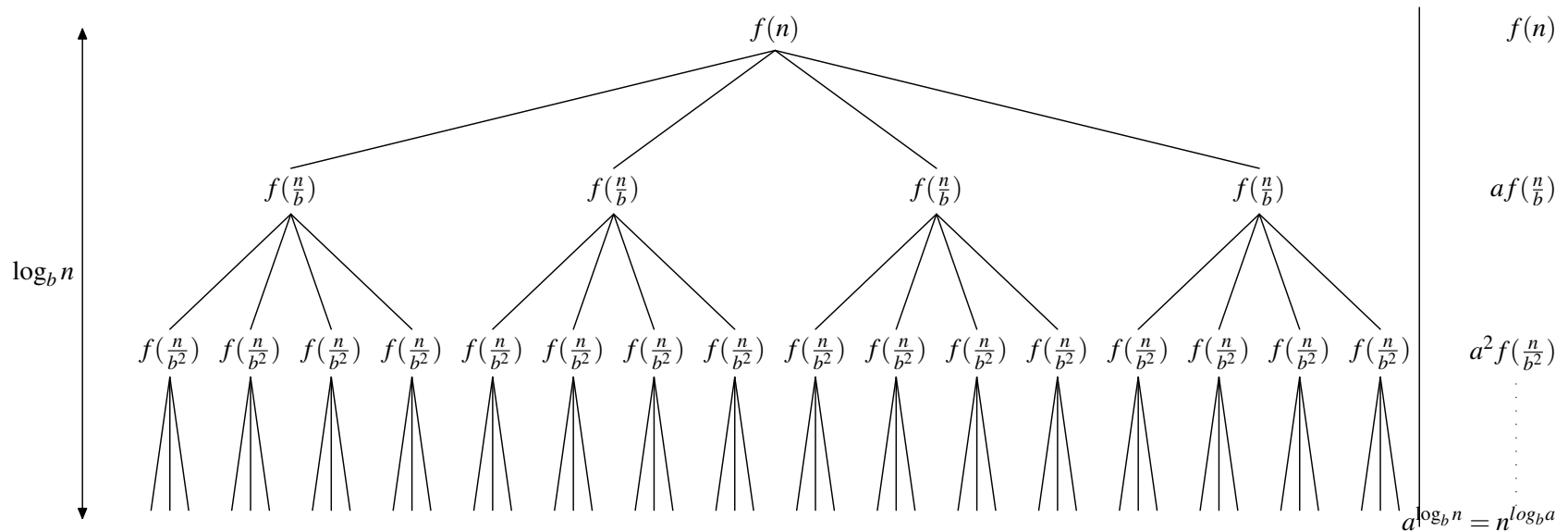
Limited Master Method

Examples:

- $T(n) = 3T(\lfloor \frac{n}{2} \rfloor) + n$: $\frac{a}{b} = \frac{3}{2} > 1$.
 $T(n) = O(n^{\log_2 3}) = O(n^{1.6})$
- $T(n) = 4T(\lfloor \frac{n}{4} \rfloor) + n$: $\frac{a}{b} = 1$. $T(n) = O(n \log n)$.
- $T(n) = 4T(\lfloor \frac{n}{5} \rfloor) + n$. $\frac{a}{b} < 1$. $T(n) = O(n)$.

Master Method

Recurrence: $T(n) = aT(n/b) + f(n)$



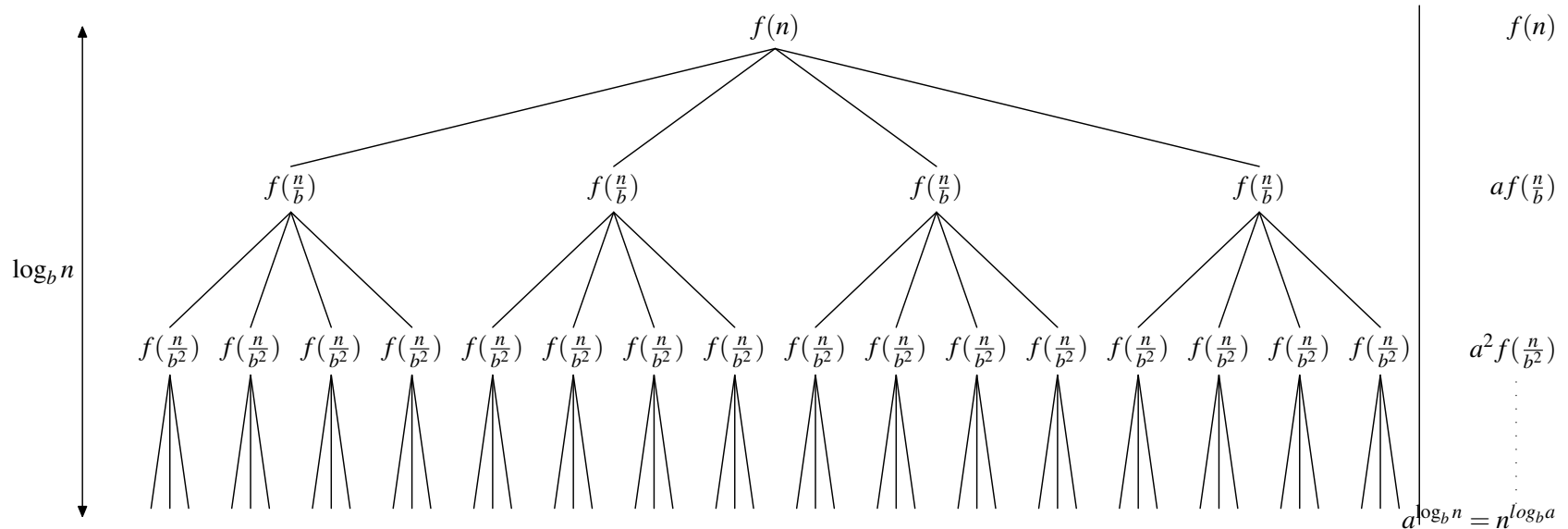
Case 1: $f(n)$ polynomially smaller than $n^{\log_b a}$ (Lots of tiny subproblems)

$$f(n) = O(n^{\log_b a - \epsilon})$$

$$T(n) = \Theta(a^{\log_b n}) = \Theta(n^{\log_b a}) = \Theta(n^{\frac{\log a}{\log b}})$$

Master Method

Recurrence: $T(n) = aT(n/b) + f(n)$

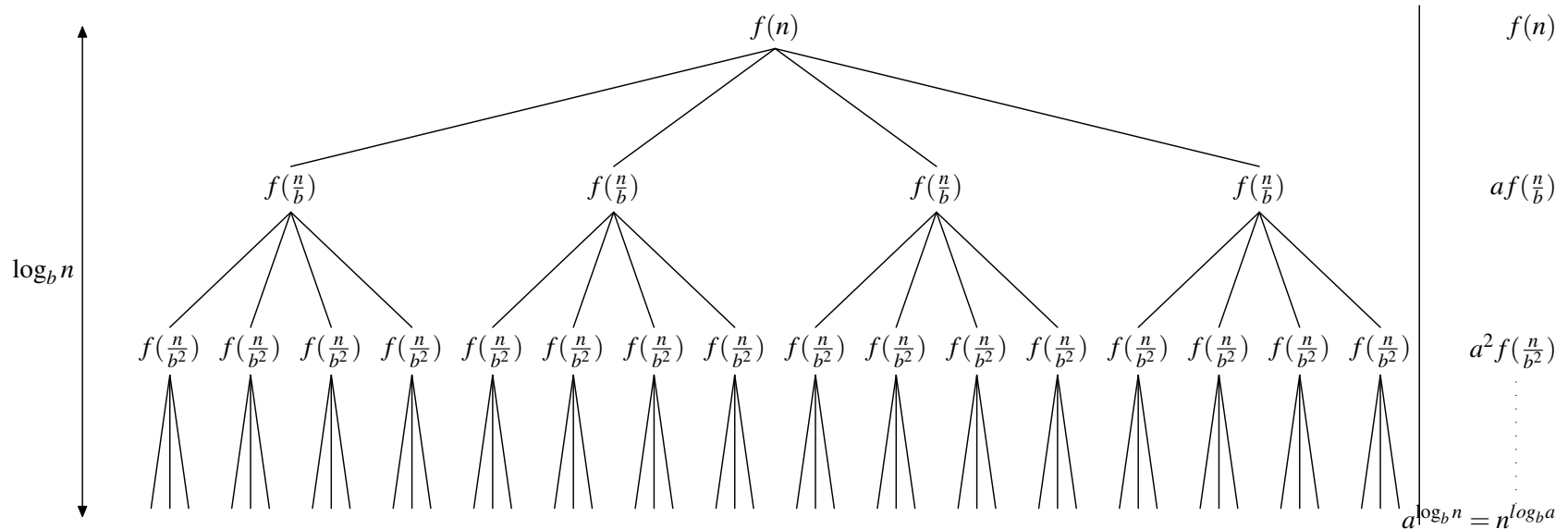


Case 2: $f(n) = \Theta(n^{\log_b a})$ (All levels about the same)

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

Master Method

Recurrence: $T(n) = aT(n/b) + f(n)$



Case 3: $f(n)$ polynomially larger than $n^{\log_b a}$ (Big step is most expensive)

$$f(n) = \Omega(n^{\log_b a + \epsilon})$$

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

Master Method

- For case 3, also need regularity: there exists $c < 1$ such that for sufficiently large n : $cf(n) \geq af(\frac{n}{b})$. That is, each lower level costs less than the previous higher level. Actually, this regularity implies polynomially larger, so case 3 need meet only this requirement.

Polynomials or $f(n) = cn^k \log^j n$ for $c, k, j \geq 0$ are regular.

- Book shows how floors and ceilings within call to T can be ignored in order analysis

$$T(n) = T(\lfloor \frac{n}{b} \rfloor), \quad S(n) = T(\lceil \frac{n}{b} \rceil), \quad R(n) = T(\frac{n}{b})$$

$$T(n) = \Theta(S(n)) = \Theta(R(n))$$

Master Method

Examples:

- $T(n) = 3T(\frac{n}{2}) + O(n^2)$: $f(n) = \Omega(n^{\log_2 3 + .2})$.
 $T(n) = O(n^2)$
- $T(n) = 3T(\frac{n}{2}) + n^{1.4}$: $f(n) = O(n^{\log_2 3 - .1})$.
 $T(n) = O(n^{\log_2 3})$.
- $T(n) = 16T(\frac{n}{4}) + 10n^2 + 3n + 5$:
 $f(n) = \Theta(n^{\log_4 16})$. $T(n) = O(n^2 \log n)$.

Master Method

Limitations:

- Non-equal partitions: $T(n) = T(\frac{4n}{7}) + T(\frac{3n}{7})$.
- Recurrences slowing down by a constant, rather than constant fraction: $T(n) = aT(n - b) + f(n)$.

- Functions in polynomial gap:

$$T(n) = 8T(n/4) + n^{1.5} \log n$$

Since $\log n < n^\epsilon$ (for sufficiently large n)

$$f(n) = n^{1.5} \log n < n^{1.5+\epsilon} \geq n^{\log_4 8} = n^{\frac{\log 8}{\log 4}} = n^{1.5}$$

- Functions failing regularity:

$$T(n) = T(\frac{n}{2}) + 4^{\lceil \log_4 n \rceil}$$