Divide and conquer
(master theorem)

Reading: Kleinberg & Tardos
Ch. 5.2
CLRS Ch. 4.5

Useful facts

- Geometric sum: when $r \neq 1$

$$\sum_{i=0}^{d} r^i = 1 + r + r^2 + \cdots + r^d = \frac{1 - r^{d+1}}{1 - r}$$

- Change of base for logs:

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

Log “flipping”:

$$\log_x y = \frac{1}{\log_y x}$$

- Exponent “power” rule

$$x^{ab} = (x^a)^b$$

$$2^{3 \cdot 2} = (2^3)^2$$

Master theorem

Consider the general recurrence:

$$T(n) = aT(\frac{n}{b}) + f(n)$$

assume O(1) base case

What?

Change base:

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

1. We know:

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

"Flip" the log:

$$\log_x y = \frac{1}{\log_y x}$$

$$\log_b n = (\log_a n)(\log_b a)$$

2. So:

$$\begin{aligned} a^{\log_b n} &= a^{(\log_a n)(\log_b a)} \\ &= \left(a^{(\log_a n)}\right)^{(\log_b a)} \\ &= n^{(\log_b a)} \end{aligned}$$

Exponent "power" rule:

$$x^{ab} = (x^a)^b$$

$$\boxed{a^{\log_b n} = n^{(\log_b a)}}$$

Back to recurrence C: T(n) = aT(n/b) + O(n), a > b

$$
\begin{aligned}
cn \sum_{i=0}^{\log_b n} (\frac{a}{b})^i &= cn \cdot \frac{1-(\frac{a}{b})^{\log_b n+1}}{1-\frac{a}{b}} && (1)\\
&= cn \cdot \frac{(\frac{a}{b})^{\log_b n+1}-1}{\frac{a}{b}-1} && (2)\\
&= \frac{cn}{\frac{a}{b}-1} \cdot \left[(\frac{a}{b})^{\log_b n+1}-1\right] && (3)\\
&\leq \frac{cn}{\frac{a}{b}-1} \cdot \left[(\frac{a}{b})^{\log_b n+1}\right] && (4)\\
&= \frac{cn}{\frac{a}{b}-1} \cdot \frac{a}{b} \cdot \left[(\frac{a}{b})^{\log_b n}\right] && (5)\\
&= \frac{ac}{b(\frac{a}{b}-1)} \cdot n \cdot \left[(\frac{a}{b})^{\log_b n}\right] && (6)
\end{aligned}
$$

Geometric sum

$$\sum_{i=0}^{d} r^i = \frac{1-r^{d+1}}{1-r}$$

Back to recurrence C: T(n) = aT(n/b) + O(n), a > b

$$= \frac{ac}{b(\frac{a}{b}-1)} \cdot n \cdot \left[(\frac{a}{b})^{\log_b n} \right] \tag{6}$$

$$= c'n \cdot \left[(\frac{a}{b})^{\log_b n} \right] \quad \text{for some constant } c' \tag{7}$$

$$= c'n \cdot \left[n^{\log_b \frac{a}{b}} \right] \tag{8}$$

$$= c'n \cdot \left[n^{\log_b a - \log_b b} \right] \tag{9}$$

$$= c'n \cdot \left[n^{\log_b a - 1} \right] \tag{10}$$

$$= c' \left[n^{1 + \log_b a - 1} \right] \tag{11}$$

$$= c' \left[n^{\log_b a} \right] \tag{12}$$

$$= O(n^{\log_b a}) \tag{13}$$

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

c' > 0 since a > b

Master theorem

Consider the general recurrence:

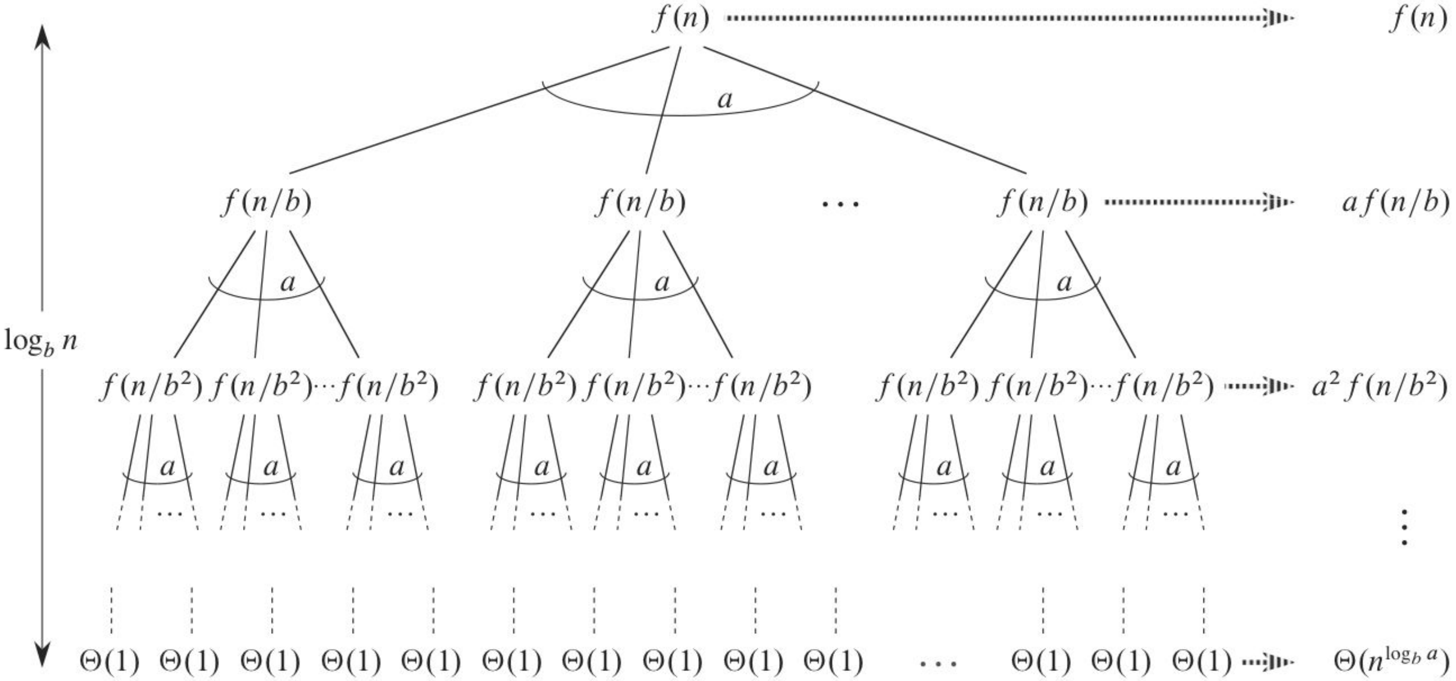
$$T(n) = aT(\frac{n}{b}) + f(n)$$

assume O(1) base case

- # leaves gives total amount of work done by base cases
- What about work outside of recursive calls?
  - How much work is done outside the recursive calls?

Master theorem

$$T(n) = aT(\frac{n}{b}) + f(n)$$



CLRS Figure 4.7

Master theorem

Consider the general recurrence:

$$T(n) = aT(\frac{n}{b}) + f(n)$$

assume O(1) base case

- # leaves gives total amount of work done by base cases
- What about work outside of recursive calls?
  - How much work is done at depth *i* outside the recursive calls?
    a^i nodes, each node is of size $\frac{n}{b^i}$
    => $a^i f(\frac{n}{b^i})$

Master theorem

Consider the general recurrence:

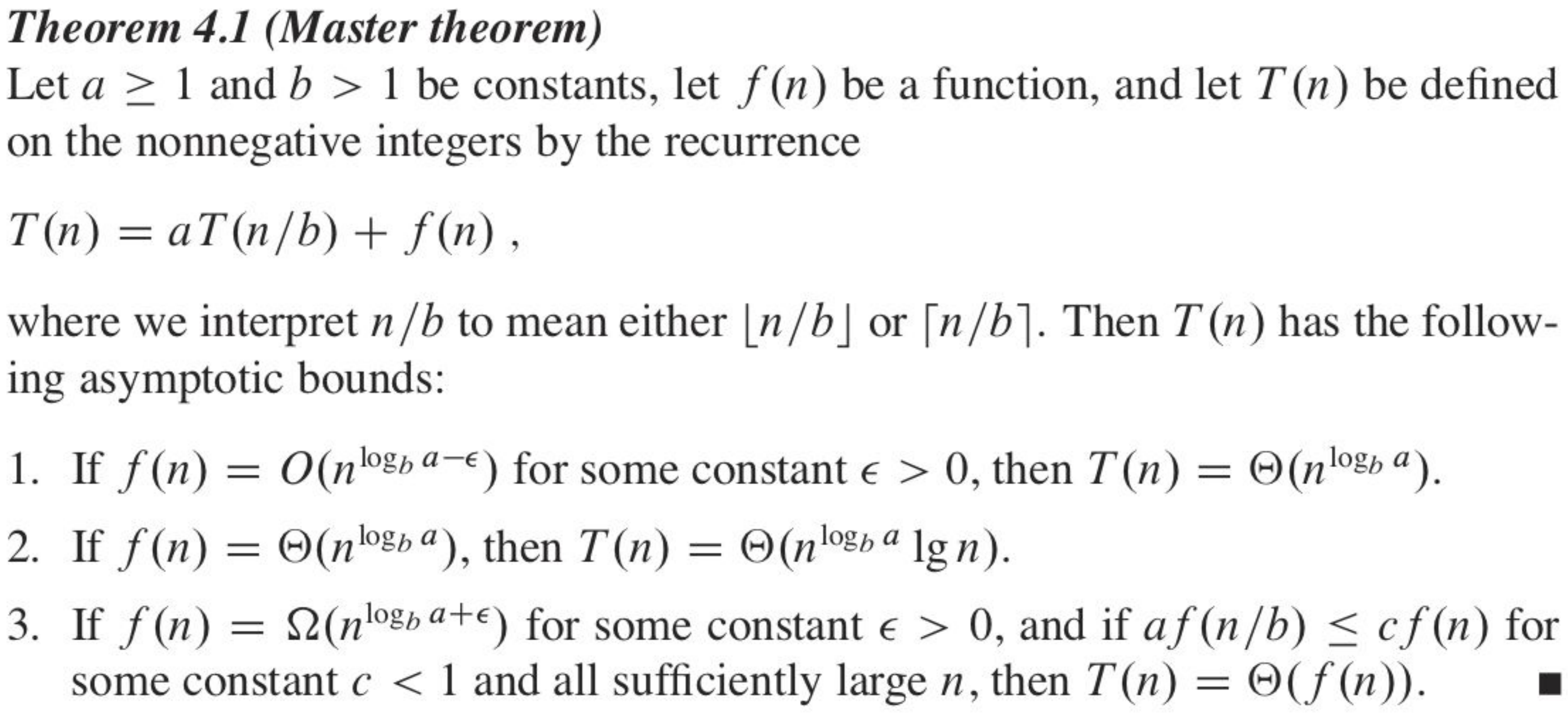
$$T(n) = aT(\frac{n}{b}) + f(n)$$

assume O(1) base case

- # leaves gives total amount of work done by base cases
- What about work outside of recursive calls?
  - depth *i:* $a^i f(\frac{n}{b^i})$

- Total is: $\underbrace{\Theta(n^{\log_b a})}_{\text{leaves/base cases}} + \underbrace{\sum_{i=0}^{\log_b n - 1} a^i f(\frac{n}{b^i})}_{\text{internal nodes - outside recursive calls}}$

Master theorem



Theorem 4.1 (Master 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

$$T(n) = aT(n/b) + f(n) ,$$

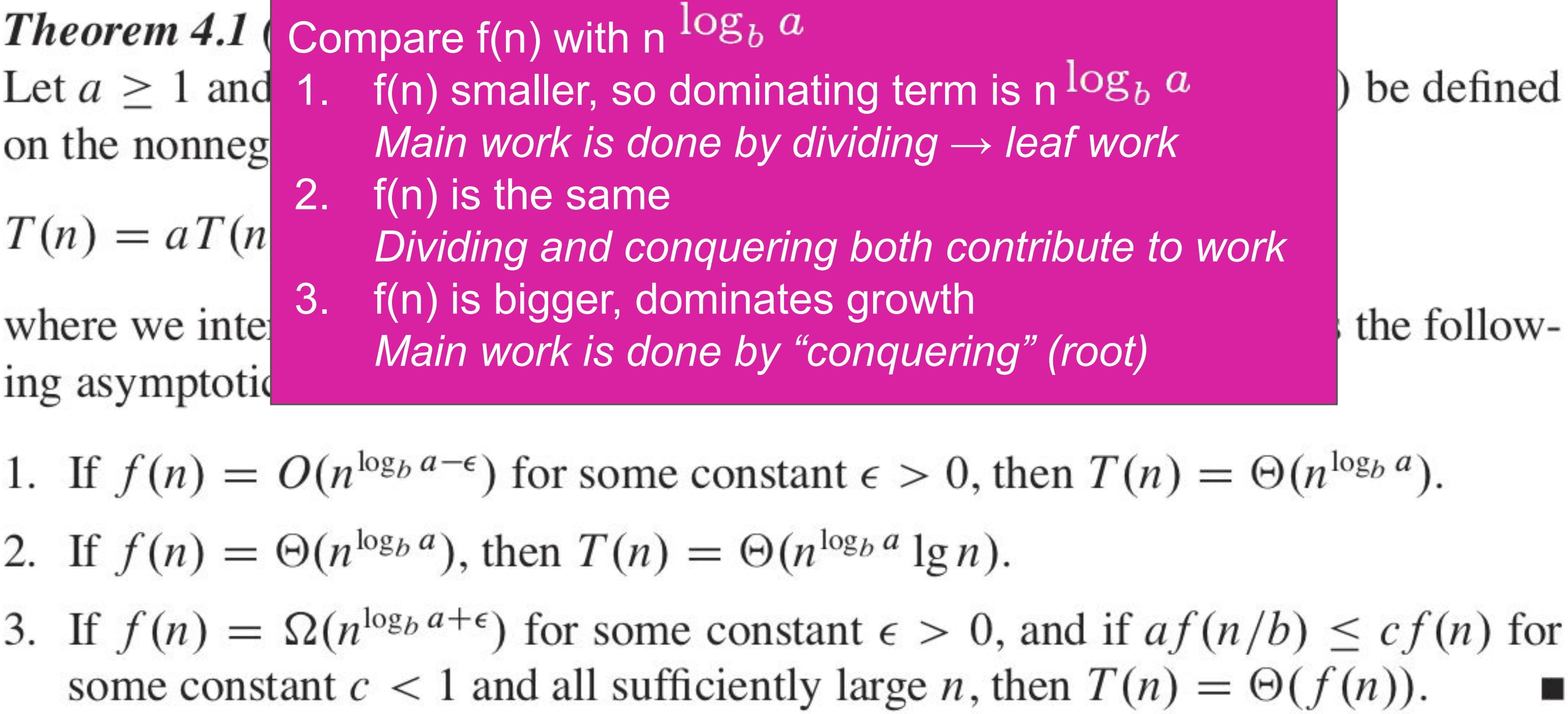
where we interpret n/b to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ has the following asymptotic bounds:

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

Master theorem

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

1. f(n) smaller, so dominating term is $n^{\log_b a}$
   Main work is done by dividing → leaf work
2. f(n) is the same
   Dividing and conquering both contribute to work
3. f(n) is bigger, dominates growth
   Main work is done by "conquering" (root)



Theorem 4.1 (

Let $a \geq 1$ and) be defined on the nonneg

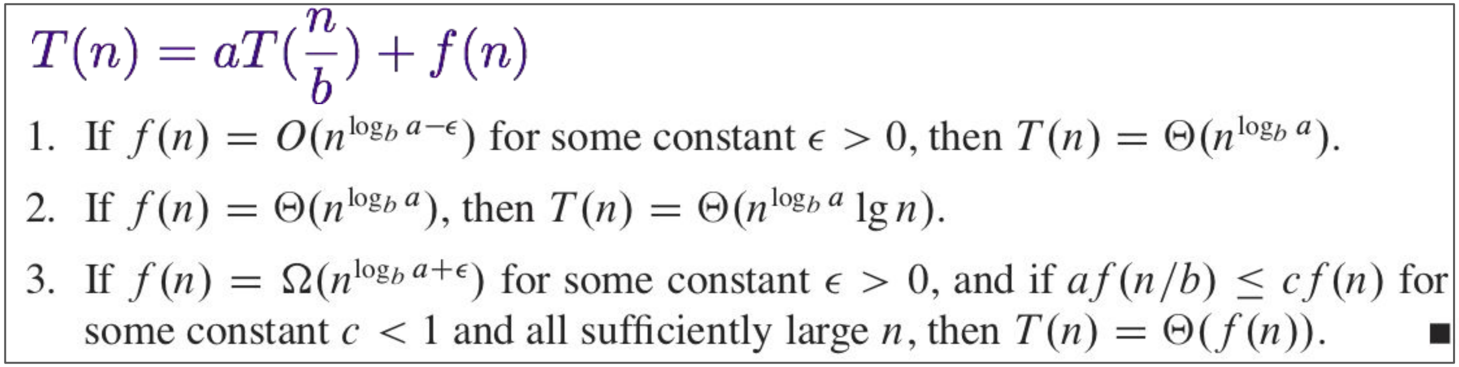
$T(n) = aT(n$

where we inte the following asymptoti

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

Huh?

Example 1



$T(n) = aT(\frac{n}{b}) + f(n)$

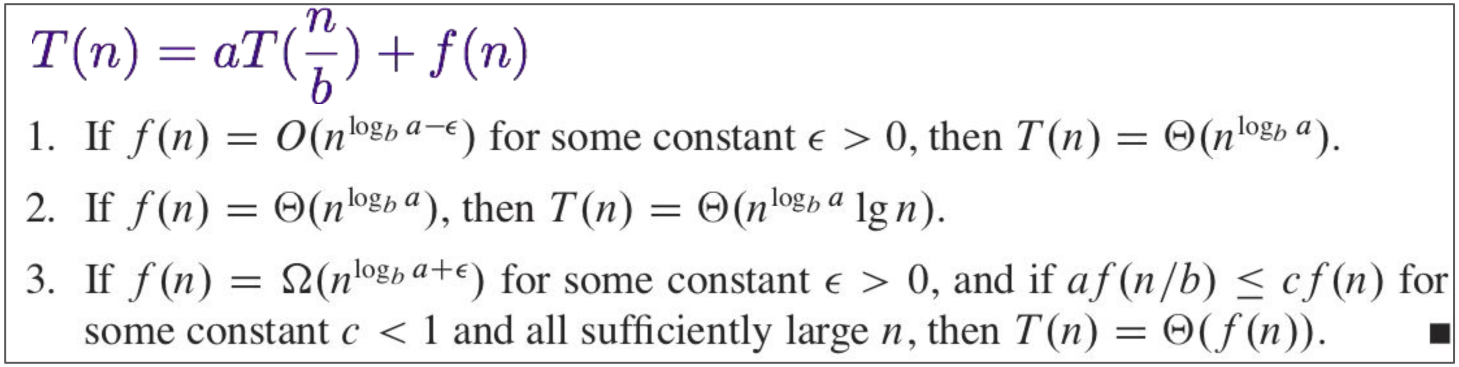
1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for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

T(n) = T(n/2) + *d*, for some constant *d*

1. Determine parameter values: a = 1, b = 2, f(n) = d = O(1)
2. Compute $\log_b a = \log_2 1 = 0$
3. Compare f(n) with $n^{\log_b a}$

   Compare O(1) with n^0=1 → tight bound Θ!
4. f(n) = Θ(1), so we are in case 2
5. Thus, T(n) = Θ(log n)

Example 2



$$T(n) = aT(\frac{n}{b}) + f(n)$$

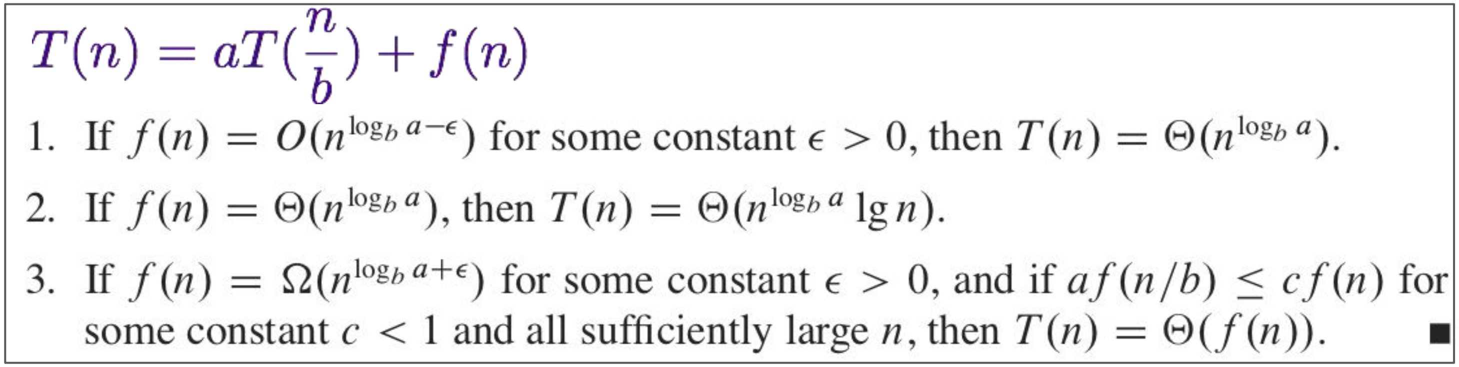
1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for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

T(n) = 9T(n/3) + n

1. Determine parameter values: a = 9, b = 3, f(n) = n
2. Compute $\log_b a = \log_3 9 = 2$
3. Compare f(n) with $n^{\log_b a}$

   Compare n with n^2; not tight bound Θ
4. f(n) = n = $O(n^2)$, so letting ϵ = 1 shows that we are in case 1
5. Thus, T(n) = $\Theta(n^2)$

Example 3



$$T(n) = aT(\frac{n}{b}) + f(n)$$

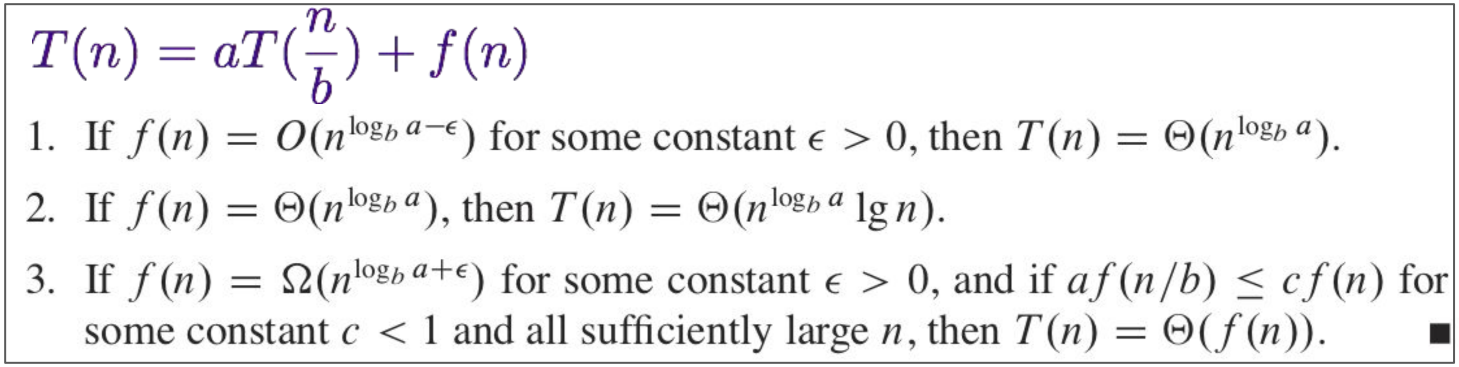
1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for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

T(n) = 3T(n/9) + 2n

1. Determine parameter values: a = 3, b = 9, f(n) = 2n
2. Compute $\log_b a = \log_9 3 = .5$
3. Compare f(n) with $n^{\log_b a}$

   Compare 2n with √n; not tight bound
4. Case 3:
   a. f(n) = 2n = Ω(n) [let ϵ = .5 > 0]
   b. 3 f(n/9) = 3 (2n/9) = 1/3 (2n) ≤ c f(n) [let c = 1/3 < 1]
5. Thus, T(n) = Θ(n)

Example 4



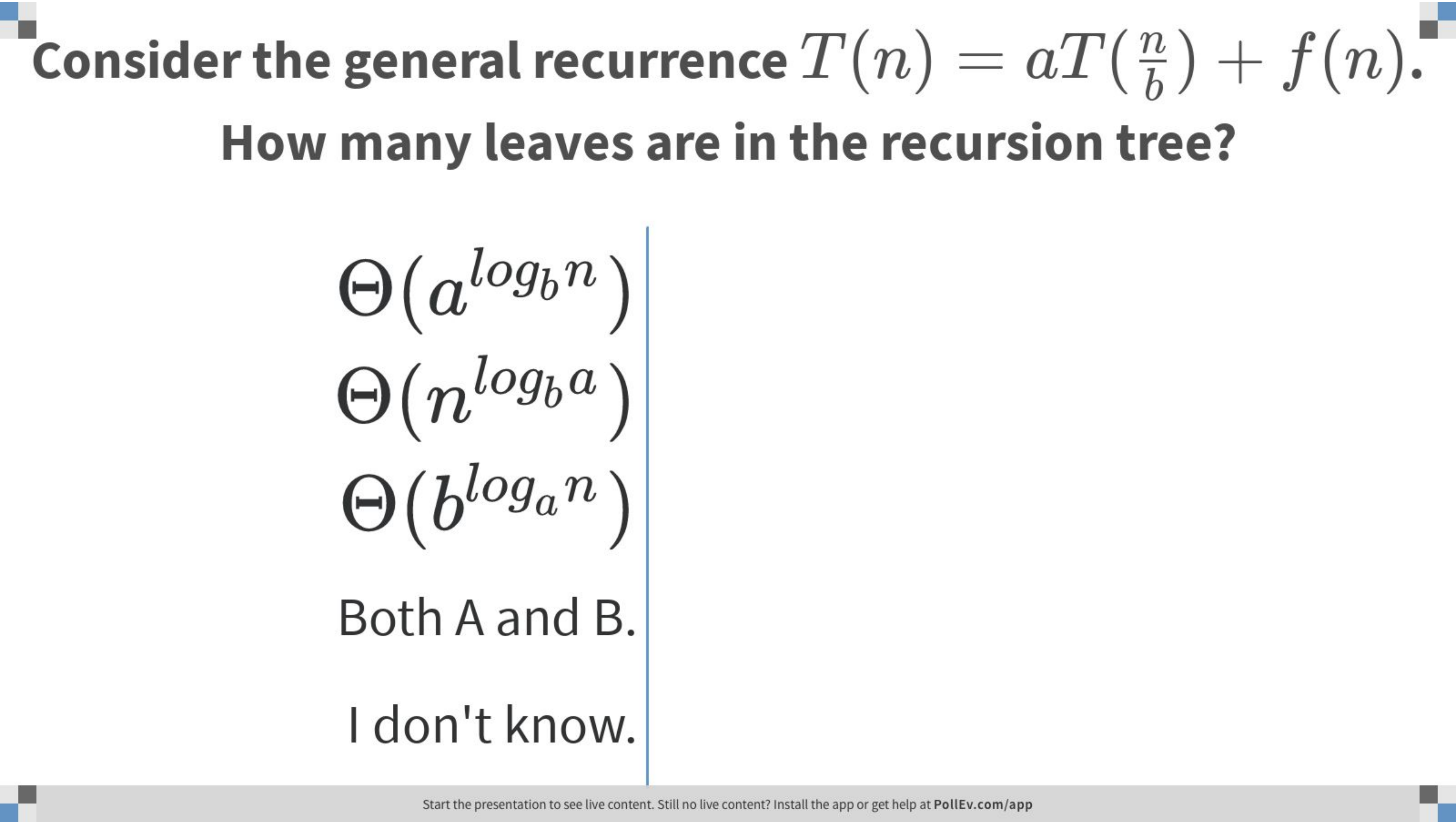
$$T(n) = aT(\frac{n}{b}) + f(n)$$

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

$T(n) = 3T(n/2) + n$

1. Determine parameter values: $a = 3, b = 2, f(n) = n$
2. Compute $\log_b a = \log_2 3 = 1.5849\ldots.$
3. Compare f(n) with $n^{\log_b a}$

   Compare n with $n^{1.585}$; not tight bound
4. Case 1: $f(n) = n = O(n)$ [let $\epsilon = .5 > 0$]
5. Thus, $T(n) = \Theta(n^{1.5849\ldots})$



Consider the general recurrence $T(n) = aT(\frac{n}{b}) + f(n)$.

How many leaves are in the recursion tree?

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

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

$\Theta(b^{log_a n})$

Both A and B.

I don't know.

Start the presentation to see live content. Still no live content? Install the app or get help at **PollEv.com/app**