# CS 4800: Algorithms & Data

Lecture 4

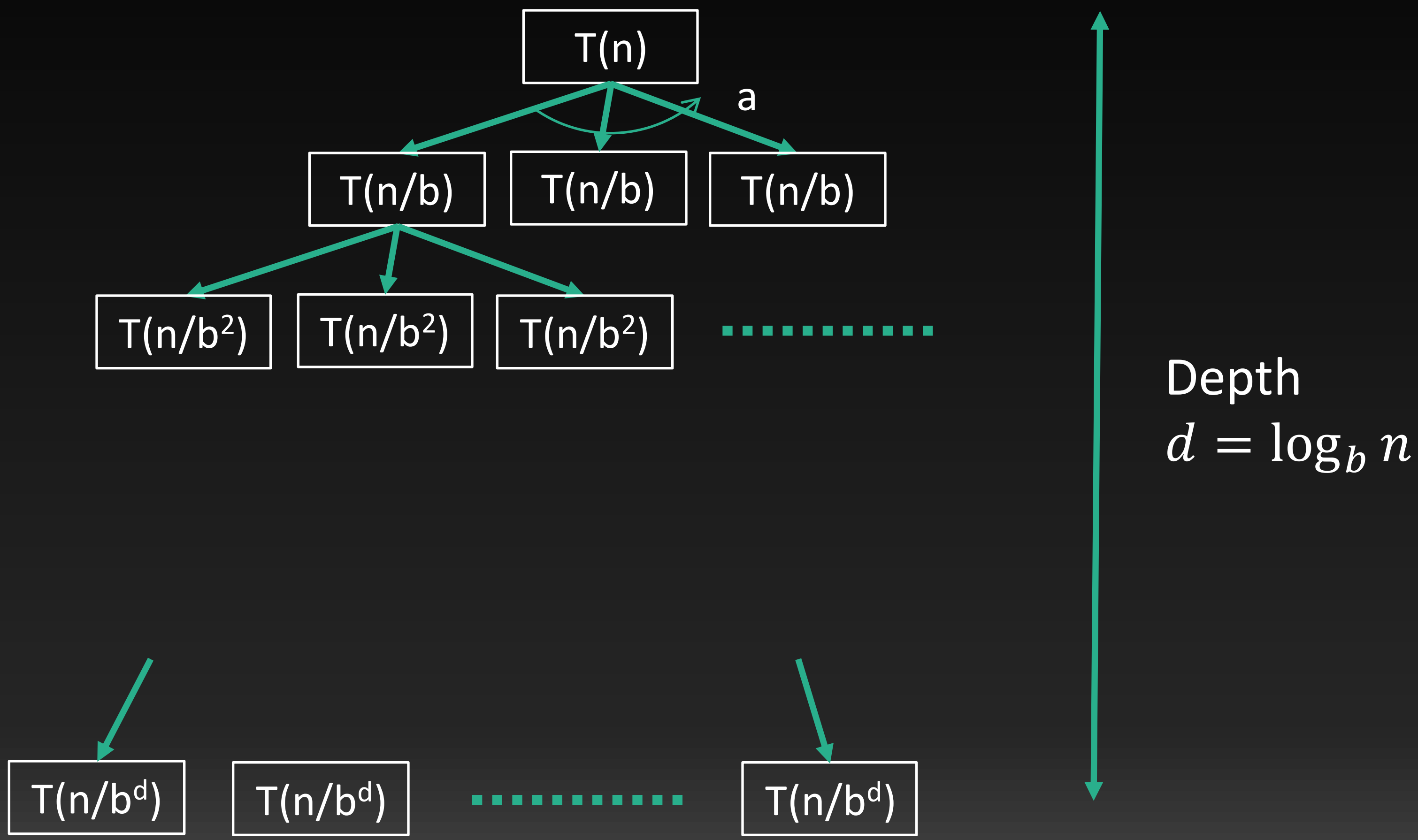
January 19, 2018

One theorem to rule them all

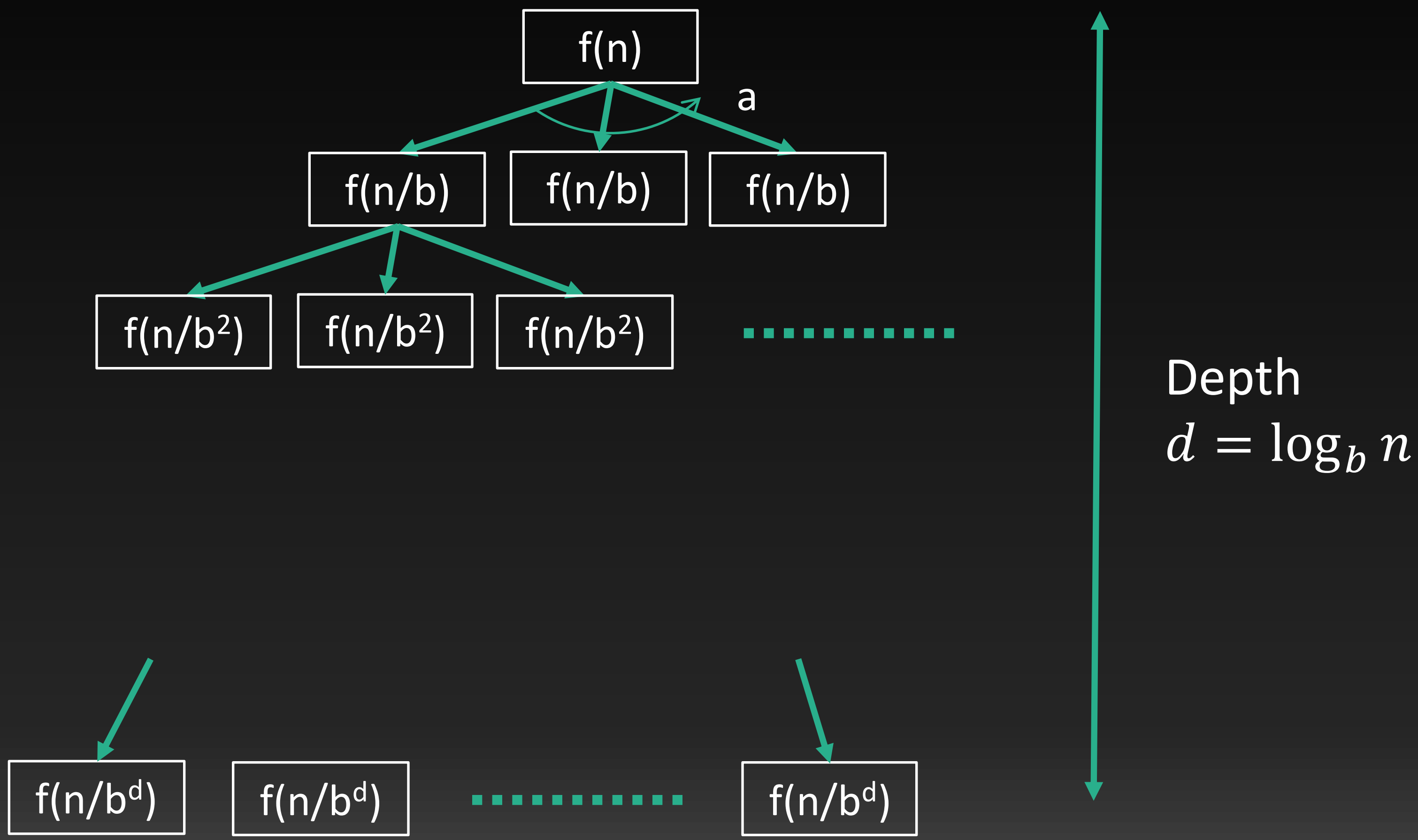
# Master theorem

$$T(n) = a \cdot T(n/b) + f(n)$$

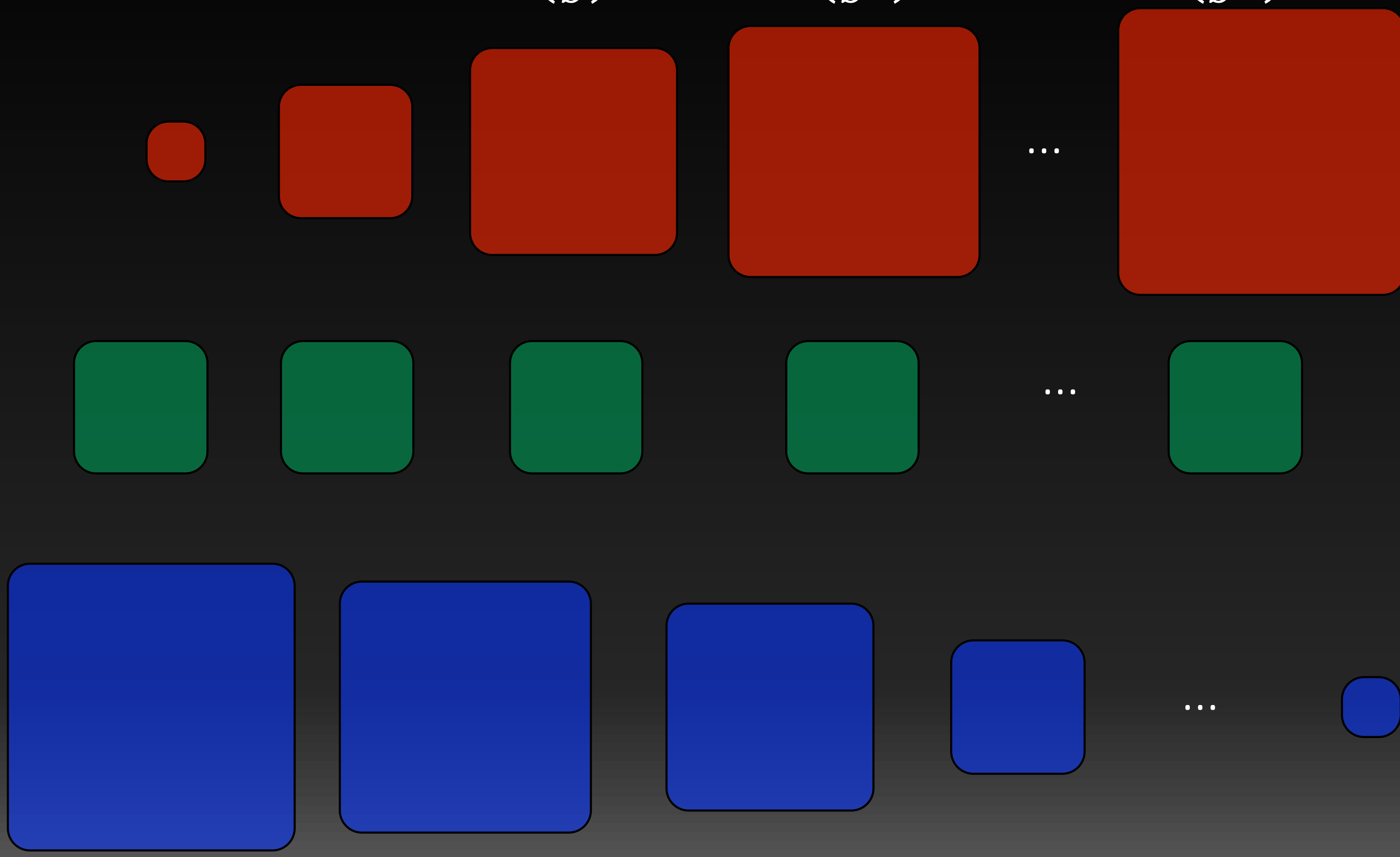
$$T(n) = a \cdot T(n/b) + f(n)$$



$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + \ldots + a^d f\left(\frac{n}{b^d}\right)$$



$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + \ldots + a^d f\left(\frac{n}{b^d}\right)$$



$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + \ldots + a^d f\left(\frac{n}{b^d}\right)$$

Case 1:

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

Example:

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

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + \ldots + a^d f\left(\frac{n}{b^d}\right)$$

Case 1: $f(n) \leq c \cdot n^{\log_b a - \epsilon}$

We have:

$$T(n) \leq cn^{\log_b a - \epsilon} + ca\left(\frac{n}{b}\right)^{\log_b a - \epsilon} + \cdots + ca^d\left(\frac{n}{b^d}\right)^{\log_b a - \epsilon}$$

$$T(n) \leq cn^{\log_b a - \epsilon}\left[1 + \frac{a}{b^{\log_b a - \epsilon}} + \frac{a^2}{(b^2)^{\log_b a - \epsilon}} + \cdots + \frac{a^d}{(b^d)^{\log_b a - \epsilon}}\right]$$

$$T(n) \leq cn^{\log_b a - \epsilon}[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{\epsilon d}]$$

$$T(n) \leq cn^{\log_b a - \epsilon}\frac{b^{\epsilon(\log_b n + 1)} - 1}{b^{\epsilon} - 1}$$

$$T(n) \leq cn^{\log_b a - \epsilon} \cdot O(n^{\epsilon} - 1) = O\left(n^{\log_b a}\right)$$

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

Case 1: Lower bound

We have:

$$T(n) \geq a^d f\left(\frac{n}{b^d}\right)$$

$$\geq a^{\log_b n} = n^{\log_b a}$$

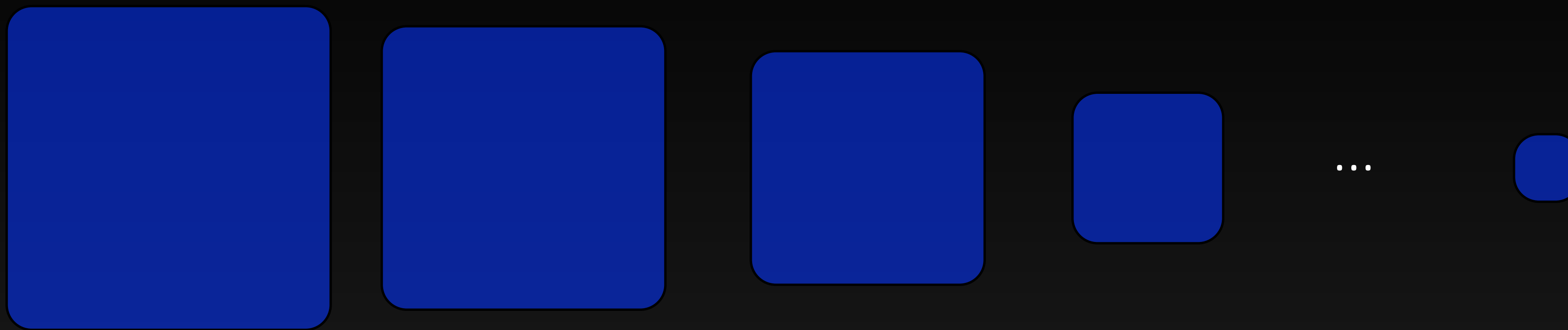
$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + \ldots + a^d f\left(\frac{n}{b^d}\right)$$

Case 2: $$f(n) = \Theta(n^{\log_b a})$$

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

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

$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + \ldots + a^d f\left(\frac{n}{b^d}\right)$$



Case 3:

$$cf(n) > a\, f\left(\frac{n}{b}\right) \text{ for c < 1, suff. large n}$$

$$\mathrm{T}(n) = \Theta(f(n))$$

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

$$T(n) = a \cdot T(n/b) + f(n)$$

case 1: 

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

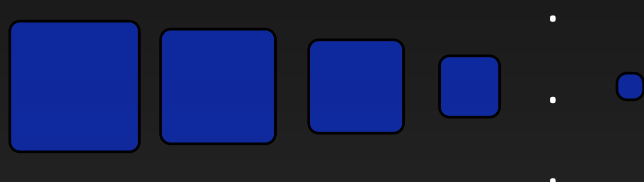
Then:

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

case 2: 

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

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

case 3: 

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

$$\mathrm{T}(n) = \Theta(f(n))$$

and c<1 s.t $c \cdot f(n) > a\, f\left(\frac{n}{b}\right)$

$$T(n) = a \cdot T(n/b) + f(n)$$

case 1:

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

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

case 2:

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

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

case 3:

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

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

and c<1 s.t $c \cdot f(n) > a\, f\left(\frac{n}{b}\right)$

1. $T(n) = 8T\left(\frac{n}{2}\right) + \Theta(n^2)$
2. $T(n) = 3T\left(\frac{n}{2}\right) + \Theta(n)$
3. $T(n) = T\left(\frac{7n}{9}\right) + 15$
4. $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n^3)$

1. $T(n) = \Theta(n^3)$
2. $T(n) = \Theta(n^{\log_2 3})$
3. $T(n) = \Theta(\log n)$
4. $T(n) = \Theta(n^3)$