the MASTER METHOD

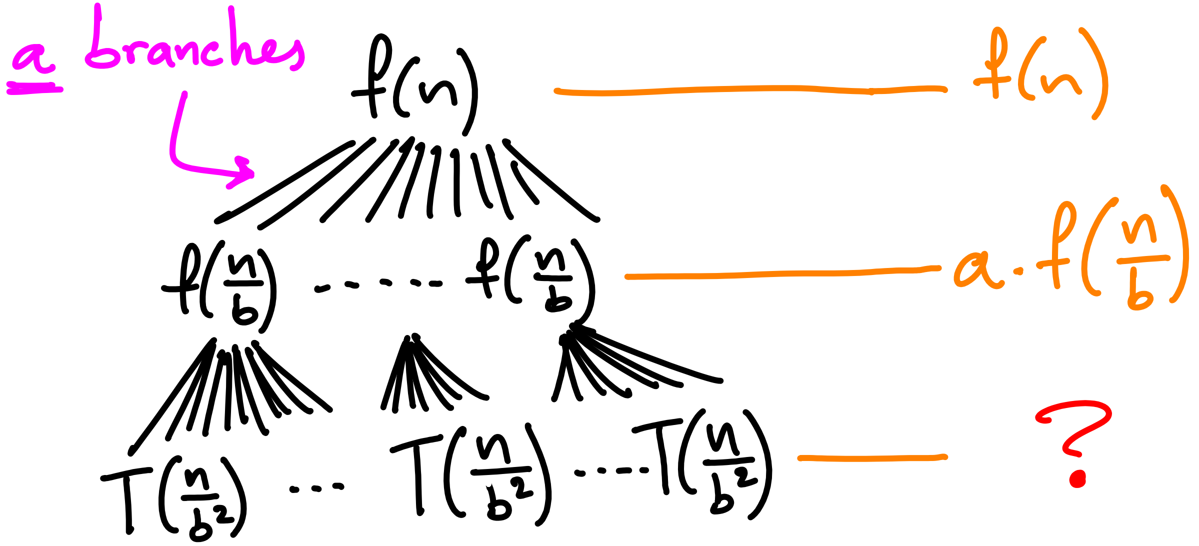
↳ a tool for solving recurrences of this form:

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

assume $\begin{cases} a \geq 1 \\ b > 1 \quad \text{otherwise you get } \infty \\ f(n) > 0 \text{ for } n > n_0 \end{cases}$

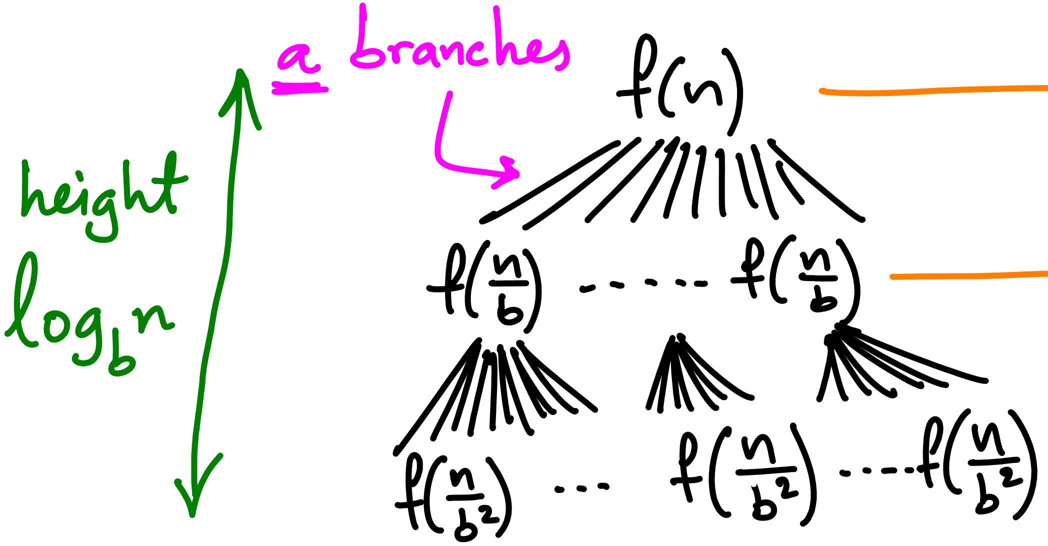
MASTER METHOD

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



MASTER METHOD

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



$f(n)$

$a \cdot f\left(\frac{n}{b}\right)$

$a^2 \cdot f\left(\frac{n}{b^2}\right)$

$\vdots$

$a^i \cdot f\left(\frac{n}{b^i}\right)$

CASE 3: root level "dominates"

CASE 2: all levels $\approx$ same

CASE 1: leaf level "dominates"

leaf level:

$$\#\text{leaves} = a^h = a^{\log_b n} = n^{\log_b a} \Big\} \text{———} \Theta\left(n^{\log_b a}\right)$$

MASTER METHOD $T(n) = aT\left(\frac{n}{b}\right) + f(n)$ compare $f(n)$ to $n^{\log_b a}$

1) $f(n) = O\left(n^{(\log_b a)-\varepsilon}\right)$

$\#\text{leaves} = \Omega(f(n)\cdot n^{\varepsilon})$

$\Leftrightarrow f(n) = O(\#\text{leaves}/n^{\varepsilon})$

leaf level dominates polynomially

2) $f(n)$

$= \Theta(\#\text{leaves})$

all levels ~ same

$\boxed{\varepsilon > 0}$

MASTER METHOD $T(n) = aT\left(\frac{n}{b}\right) + f(n)$ compare $f(n)$ to $n^{\log_b a}$

---

1) $f(n) = O\left(n^{(\log_b a) - \varepsilon}\right)$ $\quad \#\text{leaves} = \Omega\left(f(n) \cdot n^{\varepsilon}\right)$

$\quad f(n) = O\left(\#\text{leaves} / n^{\varepsilon}\right)$

leaf level dominates polynomially

---

2) $f(n) = \Theta\left(\#\text{leaves} \cdot \log^k n\right)$

all levels ~ same

[still considered ≈ same]

not mentioned in CLRS

$\varepsilon > 0$, $k \geq 0$

MASTER METHOD $T(n) = aT\left(\frac{n}{b}\right) + f(n)$ compare $f(n)$ to $n^{\log_b a}$

1) $f(n) = O\left(n^{(\log_b a)-\varepsilon}\right)$ $\quad$ $\#\text{leaves} = \Omega\left(f(n)\cdot n^{\varepsilon}\right)$ $\quad$ leaf level dominates polynomially

$\Leftrightarrow f(n) = O\left(\#\text{leaves}/n^{\varepsilon}\right)$

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

2) $f(n) = \Theta\left(n^{\log_b a}\cdot \log^k n\right) = \Theta\left(\#\text{leaves}\cdot \log^k n\right)$ $\quad$ all levels ~ same

$$T(n) = \Theta\left(f(n)\cdot \log n\right)$$

3) $f(n) = \Omega\left(n^{(\log_b a)+\varepsilon}\right) = \Omega\left(\#\text{leaves}\cdot n^{\varepsilon}\right)$ $\quad$ root dominates polynomially

AND $af\left(\frac{n}{b}\right) \leq \delta\cdot f(n)$ $\rightarrow$ work reduced by constant fraction in each level

$$\boxed{T(n) = \Theta\left(f(n)\right)}$$

$(\varepsilon > 0,\ k \geq 0,\ 0 < \delta < 1)$

$$T(n) = \underbrace{4}_{a}\,T\left(\frac{n}{\underbrace{2}_{b}}\right) + \underbrace{n}_{f(n)} \quad // \quad n^{\log_b a} = n^{\log_2 4} = n^2$$

$$f(n) = n = O(n) = O(n^{2-\overset{\varepsilon}{1}}) \quad : \text{case } 1$$

so $n^{\log_b a}$ dominates $f(n)$: answer $= n^{\log_b a} = \Theta(n^2)$

$$T(n) = 4T\left(\frac{n}{2}\right) + n^2 \quad // \quad n^{\log_b a} = n^2$$

$$f(n) = n^2 = \Theta(n^{\log_b a} \cdot \log^{\overset{k}{0}} n) \quad : \text{case } 2$$

ANS: $\Theta(n^{\log_b a} \cdot \log^{k+1} n) = \Theta(n^2 \log n)$

$$T(n) = 4T\left(\frac{n}{2}\right) + n^3 \quad // \quad f(n) = n^3 = \Omega(n^{2+\varepsilon}) \text{ AND } af\left(\frac{n}{b}\right) = 4\cdot\left(\frac{n}{2}\right)^3 = \frac{1}{2}n^3 = \underbrace{\frac{1}{2}}_{\delta} \cdot f(n)$$

↳ case 3 : ANS: $\Theta(n^3)$

$$T(n) = 4T\left(\frac{n}{2}\right) + \frac{n^2}{\log n} \quad \Big\} \text{ N/A}$$

But in fact there is now yet another extension of case 2

(see last page)

Divide & Conquer

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

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

if $f(n) = n \Rightarrow$ case 2 $\Rightarrow \Theta(n \cdot \log n)$

$(k=0)$

if $f(n) = n \log^k n \Rightarrow$ case 2 $\Rightarrow \Theta(n \log^{k+1} n)$

if $f(n) = \log^c n \Rightarrow$ case 1 $\Rightarrow \Theta(n)$

$f(n) = O(n^{1-\varepsilon})$

if $f(n) = n^c$ for $c > 1$ $\Big\}$ $f(n) = \Omega(n^{1+\varepsilon})$

AND $2f\left(\frac{n}{2}\right) = \frac{2}{2^c} n^c = \frac{1}{2^{c-1}} \cdot f(n)$

... case 3 $\Rightarrow \Theta(f(n))$

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

SAME

FYI - EXTRA-EXTENDED CASE 2 $\quad f(n) = \Theta(n^{\log_b a} \cdot \log^k n)$

(Doesn't come up in any algorithms that we will see)

Standard extended case 2

$k \geq 0 \quad T(n) = \Theta(n^{\log_b a} \cdot \log^{k+1} n) \qquad T(n) = \Theta(f(n) \cdot \log n)$

$k = -1 \quad T(n) = \Theta(n^{\log_b a} \cdot \log\log n) \qquad T(n) = \Theta(f(n) \cdot \log n \cdot \log\log n)$

e.g., $T(n) = 8T(\frac{n}{2}) + \frac{n^3}{\log n} = n^3 \log\log n$

$k \leq -2 \quad T(n) = \Theta(n^{\log_b a})$ almost like an extended case 1:

Leaf level dominates by a "large" poly-log factor.