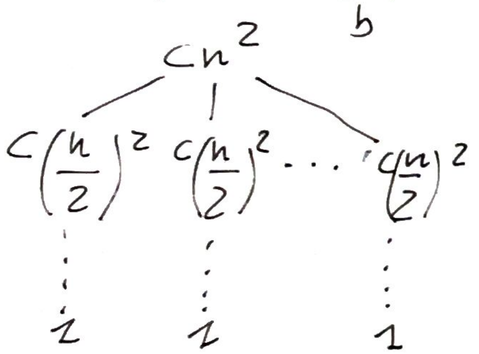
Matrix multiplication - Complexity of the recursive algorithm

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ \underset{a}{8}\, T\left(\frac{n}{\underset{b}{2}}\right) + \Theta(n^2) & \text{if } n > 1 \end{cases}$$



$\longrightarrow \frac{8}{4} n^2$

each level has 8 times the nodes of the previous level, so 8^i nodes where each node has cost $c\left(\frac{n}{2^i}\right)^2 = \frac{cn^2}{4^i}$

The total cost per level is $\left(\frac{8}{4}\right)^i cn^2 = 2^i cn^2$

$$\underbrace{\sum_{i=0}^{\log_2 n - 1} 2^i cn^2}_{\text{levels} \cdot \text{cost}} + \underbrace{n^{\log_2 8}}_{\text{leaves}}$$

$$\overset{!}{=} \sum_{i=0}^{\log_2 n - 1} 2^i cn^2 + \Theta(n^3)$$

$2^{\log_2 n} \sim n^{\log_2 2} = n$

$$\overset{!}{=} \frac{1 - 2^{\log_2 n}}{1 - 2} \cdot cn^2 + \Theta(n^3)$$

$$\overset{!}{=} -\frac{1 - n}{1} \cdot cn^2 + \Theta(n^3)$$

$$\overset{!}{=} n - 1 \cdot cn^2 + \Theta(n^3) = cn^3 - 1 + \Theta(n^3) \Rightarrow \Theta(n^3)$$

The master method for solving recurrences

$T(n) = aT\left(\frac{n}{b}\right) + f(n)$, where $a \geq 1$ and $b > 1$ and $f(n)$ is asymtotically positive.

$\boxed{\text{Theorem}}$

Let $a \geq 1$ and $b > 1$ be constants, and let $f(n)$ be a function. Let $T(n)$ be defined on the nonnegative integers by the recurrence

$T(n) = aT\left(\frac{n}{b}\right) + f(n)$, where $\frac{n}{b}$ could be either $\lfloor \frac{n}{b} \rfloor$ or $\lceil \frac{n}{b} \rceil$. Then, $T(n)$ has the following bounds

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

In the first case, not only $f(n)$ has to be smaller than $n^{\log_b a}$, it MUST be POLINOMIALLY smaller.

↳ $f(n)$ must be smaller than $n^{\log_b a}$ by a factor n^{ε} for $\varepsilon > 0$.

Same goes for case 3. So, there are "gaps" where the master theorem cannot be applied.

Exercises

a) $T(n) = 9T\left(\frac{n}{3}\right) + n.$

$f(n) = n \qquad n^{\log_3 9} = n^2$

$f(n) = O\left(n^{2-\varepsilon}\right)$ if $\varepsilon = 1 \Rightarrow T(n) = \Theta(n^2)$

b) $T(n) = T\left(\frac{2n}{3}\right) + 1$

$f(n) = 1 \qquad n^{\log_{3/2} 1} = 1 \Rightarrow \underline{f(n) = n^{\log_b a}}$

↳ case 2

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

c) $T(n) = 3T\left(\frac{n}{4}\right) + n \lg n$

$f(n) = n \lg n \qquad n^{\log_4 3} = n^{0.793}$

$f(n) \overset{?}{=} \Omega\left(n^{\log_4 3 + \varepsilon}\right) \xrightarrow{?} n \lg n = \Omega(n)$

↳ $\varepsilon \cong 0.2$

↓

Ok!

we need to prove the "regularity" condition

$3 \cdot \frac{n}{4} \lg \frac{n}{4} \leq c \cdot n \lg n \quad , c < 1$

$\Rightarrow \lg n > \lg \frac{n}{4}$ and if $\frac{3}{4} < c < 1$ then $\frac{3}{4} n < cn$

so "regularity" condition is verified, then $T(n) = \Theta(n \lg n)$

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

we are between case 2 and 3 $\Rightarrow$ the M.T. cannot be applied.

$f(n) = n \lg n \qquad n^{\log_2 2} = n$

It looks like $n \lg n = \Omega\left(n^{1+\varepsilon}\right)$ but is it polynomially smaller?

$n \lg n$ must be smaller than n by a factor n^{ε}

but $\frac{n \lg n}{n} = \lg n$ is not polynomially larger than n^{ε} for any $\varepsilon > 0$