L4 5800

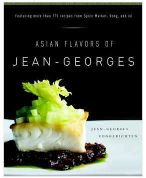
jan 28/31 2022

shelat

Announcements on H1

tree method

?-✓ guess & check (induction)

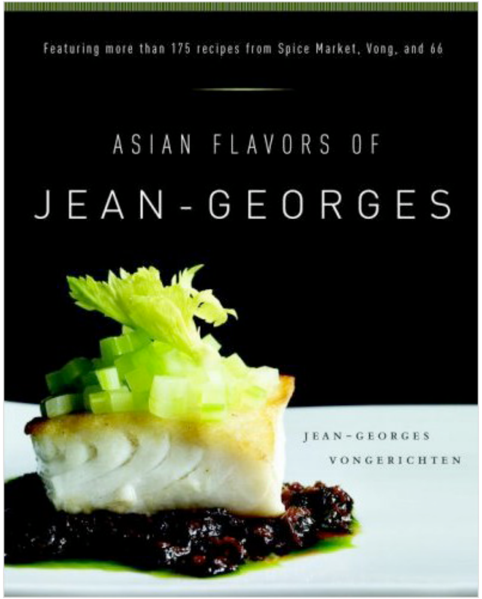


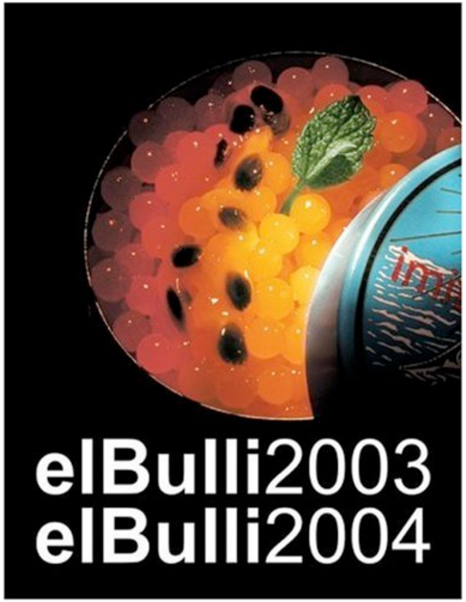
→ Master's Thm. (cookbook)

→ substitution, change of variable -

http://www.drblank.com/law301.jpg

cookbook

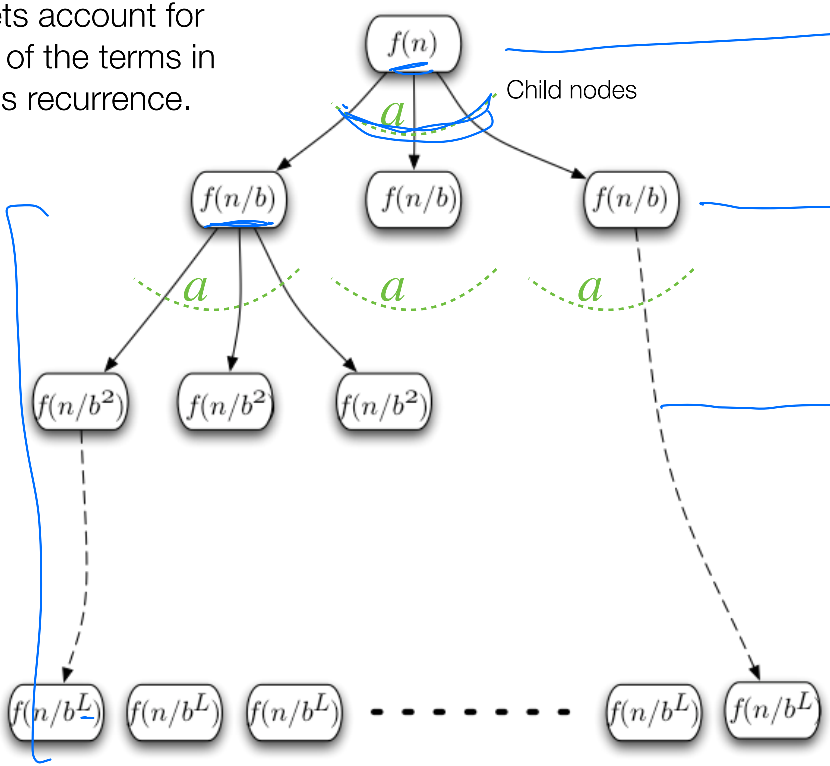

Featuring more than 175 recipes from Spice Market, Vong, and 66
ASIAN FLAVORS OF
JEAN-GEORGES
JEAN-GEORGES
VONGERICHTEN


elBulli2003
elBulli2004

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

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

Lets account for all of the terms in this recurrence.



$f(n)$

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

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

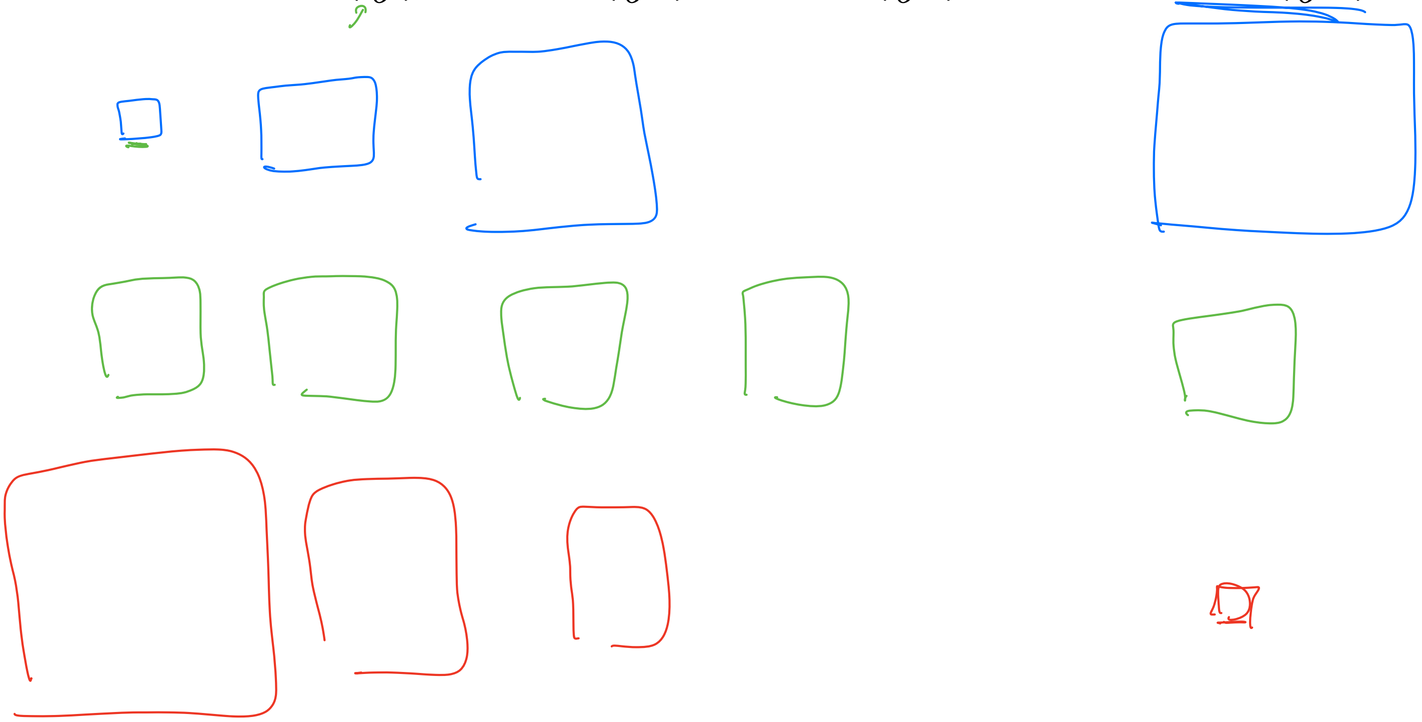
$a^L \cdot f\left(\frac{n}{b^L}\right)$

$b^L = n$

$\log_b(b^L) = \log_b n$

$L = \log_b n$

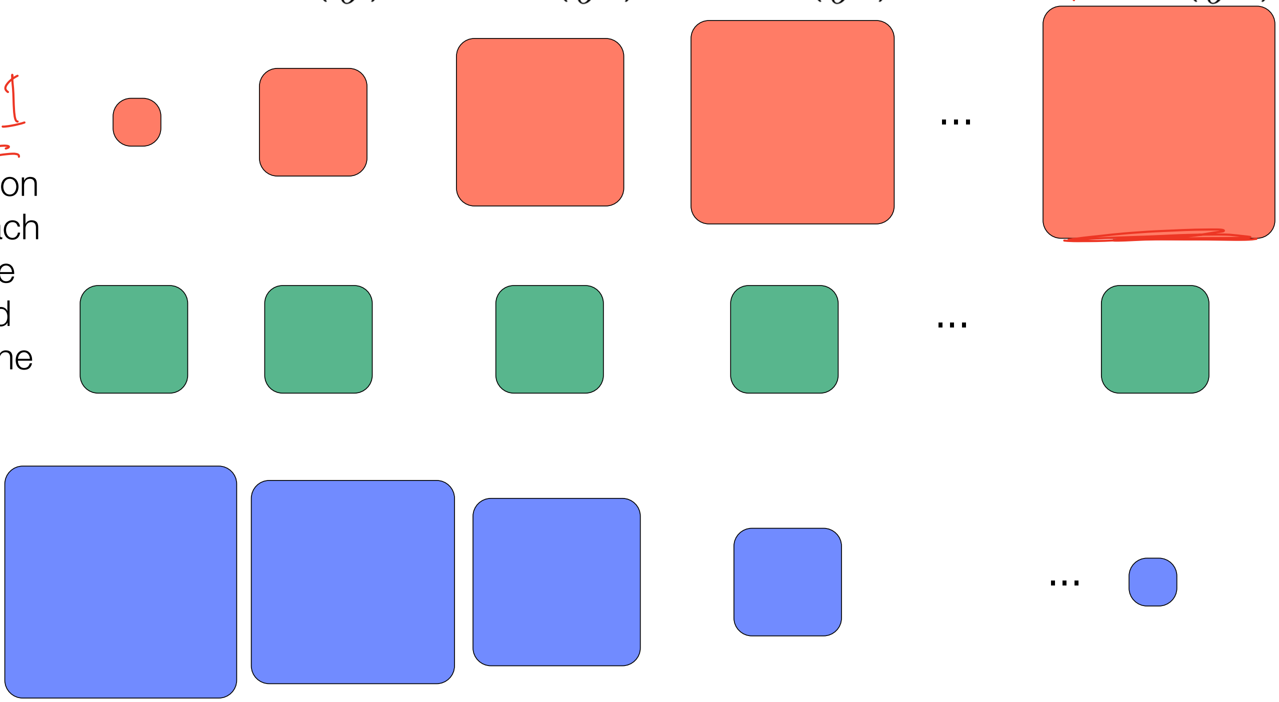
$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$



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

CASE 1

Comparison of how each term in the sum could relate to the others

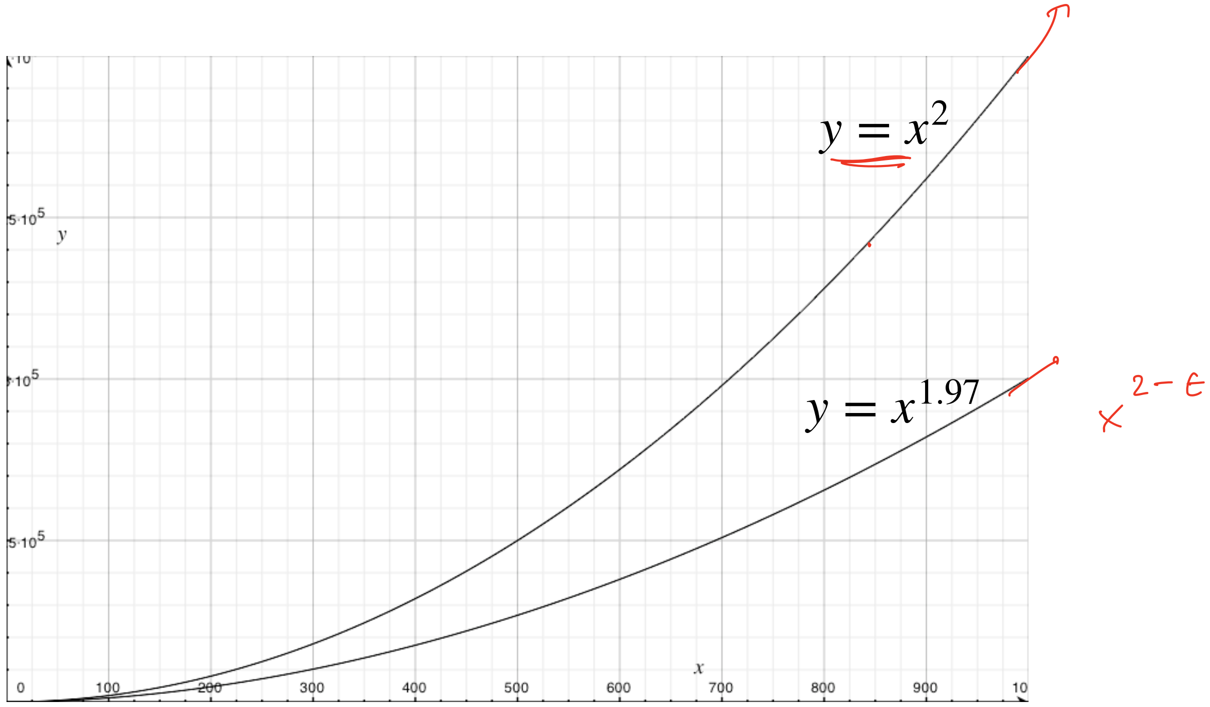


$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$$

case 1: $f(n) = O(n^{\log_b a - \epsilon})$

small positive value like 0.001

"f is asymptotically smaller than $n^{\log_b a}$"


$y = x^2$
$y = x^{1.97}$
$x^{2-\epsilon}$
y
x
0
100
200
300
400
500
600
700
800
900

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

case 1: $$f(n) = O(n^{\log_b a - \epsilon})$$

example: $T(n) = \underset{a}{4}T(n/\underset{b}{\underline{2}}) + \underline{n}$

$f(n) = n$

$$\boxed{f(n) = n} = O\left(\boxed{n^{\log_2 4 - \epsilon} = n^{2-\epsilon}}\right)$$

$\epsilon = 0.01$

So case I applies

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

case 1: $f(n) = O(n^{\log_b a - \epsilon})$

$$f(n) < c \cdot n^{\log_b a - \epsilon}$$

$$T(n) < c \cdot n^{\log_b a - \epsilon} + a \cdot c \cdot \left(\frac{n}{b}\right)^{\log_b a - \epsilon} + a^2 \cdot c \left(\frac{n}{b^2}\right)^{\log_b a - \epsilon} + \cdots + a^L \cdot c \left(\frac{n}{b^L}\right)^{\log_b a - \epsilon}$$

$$= c \cdot n^{\log_b a - \epsilon}\left[1 + \frac{a}{b^{\log_b a - \epsilon}} + \frac{a^2}{b^{2(\log_b a - \epsilon)}} + \cdots + \frac{a^L}{b^{L(\log_b a - \epsilon)}}\right]$$

Notice $b^{i(\log_b a - \epsilon)} = \frac{b^{i(\log_b a)}}{b^{i(\epsilon)}} = \frac{a^i}{b^{i \cdot \epsilon}}$

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

case 1: $f(n) = O(n^{\log_b a - \epsilon})$

$$T(n) \le cn^{\log_b a-\epsilon} + ac\left(\frac{n}{b}\right)^{\log_b a-\epsilon} + a^2c\left(\frac{n}{b^2}\right)^{\log_b a-\epsilon} + \cdots + a^Lc\left(\frac{n}{b^L}\right)^{\log_b a-\epsilon}$$

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

case 1: $f(n) = O(n^{\log_b a - \epsilon})$

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

$$= c \cdot n^{\log_b a - \epsilon}\left[1 + \frac{\cancel{a}}{\frac{\cancel{a}}{b^\epsilon}} + \frac{\cancel{a^2}}{\frac{\cancel{a^2}}{b^{2\epsilon}}} + \cdots + \frac{\cancel{a^L}}{\frac{\cancel{a^L}}{b^{L\epsilon}}}\right]$$

$$= c \cdot n^{\log_b a - \epsilon}\left[1 + b^\epsilon + b^{2\epsilon} + \cdots b^{L\epsilon}\right]$$

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

case 1: $f(n) = O(n^{\log_b a - \epsilon})$

$$T(n) \le cn^{\log_b a-\epsilon} + ac\left(\frac{n}{b}\right)^{\log_b a-\epsilon} + a^2c\left(\frac{n}{b^2}\right)^{\log_b a-\epsilon} + \cdots + a^Lc\left(\frac{n}{b^L}\right)^{\log_b a-\epsilon}$$

$$= cn^{\log_b a-\epsilon}\left[1 + \left(\frac{a}{b^{\log_b a-\epsilon}}\right) + \left(\frac{a^2}{b^{2(\log_b a-\epsilon)}}\right) + \cdots + \left(\frac{a^L}{b^{L(\log_b a-\epsilon)}}\right)\right]$$

$$= cn^{\log_b a-\epsilon}\left[1 + \left(\frac{a}{a/b^{\epsilon}}\right) + \left(\frac{a^2}{a^2/b^{2\epsilon}}\right) + \cdots + \left(\frac{a^L}{a^L/b^{L\epsilon}}\right)\right]$$

$$= cn^{\log_b a-\epsilon}\left[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{L\epsilon}\right]$$

$$= cn^{\log_b a - \epsilon}\left[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{L\epsilon}\right]$$

$$= cn^{\log_b a - \epsilon}\left[\frac{b^{\epsilon(L+1)} - 1}{b^{\epsilon} - 1}\right]$$

$$b^L = b^{\log_b n}$$

$$= c \cdot n^{\log_b a - \epsilon}\left[\frac{b^{\epsilon} \cdot n^{\epsilon} - 1}{b^{\epsilon} - 1}\right]$$

Since $b > 1, \epsilon > 0$

then $b^{\epsilon} > 1$

$$\leq c \cdot n^{\log_b a - \epsilon} \cdot n^{\epsilon} \cdot \left(\frac{b^{\epsilon}}{b^{\epsilon} - 1}\right) = O\left(n^{\log_b a}\right)$$

$$= cn^{\log_b a - \epsilon}\left[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{L\epsilon}\right]$$

$$= cn^{\log_b a - \epsilon}\left[\frac{b^{\epsilon(L+1)} - 1}{b^{\epsilon} - 1}\right]$$

Recall that
$b^L = b^{\log_b n} = n$

Since $b > 1, \epsilon > 0$
then $b^{\epsilon} > 1$

$$= cn^{\log_b a - \epsilon}\left[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{L\epsilon}\right]$$

$$= cn^{\log_b a - \epsilon}\left[\frac{b^{\epsilon(L+1)} - 1}{b^{\epsilon} - 1}\right]$$

Recall that
$b^L = b^{\log_b n} = n$

$$= cn^{\log_b a - \epsilon}\left[\frac{b^{\epsilon}n^{\epsilon} - 1}{b^{\epsilon} - 1}\right]$$

Since $b > 1, \epsilon > 0$
then $b^{\epsilon} > 1$

$$= cn^{\log_b a - \epsilon}\left[1 + b^{\epsilon} + b^{2\epsilon} + \cdots + b^{L\epsilon}\right]$$

$$= cn^{\log_b a - \epsilon}\left[\frac{b^{\epsilon(L+1)} - 1}{b^{\epsilon} - 1}\right]$$

Recall that
$b^L = b^{\log_b n} = n$

$$= cn^{\log_b a - \epsilon}\left[\frac{b^{\epsilon}n^{\epsilon} - 1}{b^{\epsilon} - 1}\right]$$

Since $b > 1, \epsilon > 0$
then $b^{\epsilon} > 1$

$$\leq \left[\frac{cb^{\epsilon}}{b^{\epsilon} - 1}\right] n^{\log_b a - \epsilon} n^{\epsilon} = O(n^{\log_b a})$$

ignore these terms

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

case 1: Lower bound

We have:

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

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

Recall that L is the depth of the recursion and

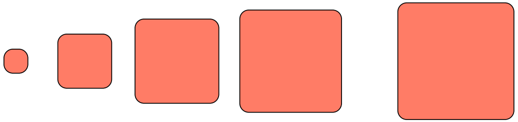
$$a^L = a^{\log_b n}$$

$$\left(\underbrace{b^{\log_b a}}_{=a}\right)^{\log_b n} = b^{(\log_b n)(\log_b a)} = n^{\log_b a}$$

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

Master's Theorem $T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$

Case 1



$f(n) = O(n^{\log_b a - \epsilon}), \epsilon > 0$

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

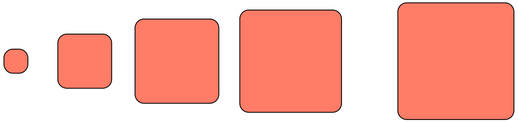
Case 2

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

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

Master's Theorem $T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$

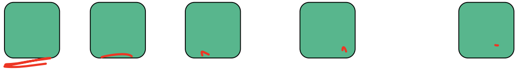
Case 1



$f(n) = O(n^{\log_b a - \epsilon}), \epsilon > 0$

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

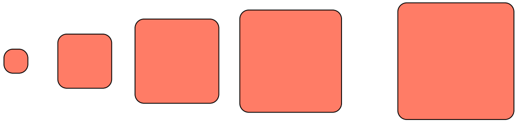
Case 2



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

Master's Theorem $T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^L f\left(\frac{n}{b^L}\right)$

Case 1



$f(n) = O(n^{\log_b a - \epsilon}), \epsilon > 0$

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

Case 2

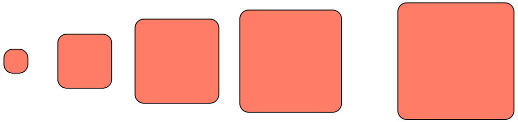
$f(n) = \Theta(n^{\log_b a})$

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

Master's Theorem

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

Case 1



$$f(n) = O(n^{\log_b a - \epsilon}), \epsilon > 0$$

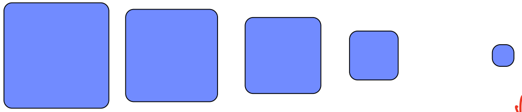
Then $T(n) = \Theta(n^{\log_b a})$

Case 2

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

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

Case 3



regularity

$f(n) = \Omega(n^{\log_b a + \epsilon}), \epsilon > 0$ and $\exists c < 1,\ af(n/b) < cf(n)$

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

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

case 2: $c'n^{\log_b a} < f(n) < cn^{\log_b a}$

$$T(n) < c \cdot n^{\log_b a} + a \cdot c \left(\frac{n}{b}\right)^{\log_b a} + a^2 \cdot c \cdot \left(\frac{n}{b^2}\right)^{\log_b a} + \cdots + a^L \cdot c \left(\frac{n}{b^L}\right)^{\log_b a}$$

$$= c \cdot n^{\log_b a}\left[1 + \frac{a}{b^{\log_b a}} + \frac{a^2}{b^{2 \cdot \log_b a}} + \cdots + \frac{a^L}{b^{L \cdot \log_b a}}\right]$$

$$b^{i \cdot \log_b a} = \underline{a^i}$$

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

case 2: $c' n^{\log_b a} < f(n) < c n^{\log_b a}$

$$T(n) < c n^{\log_b a} \left[1 + \left(\frac{a}{b^{\log_b a}} \right) + \left(\frac{a^2}{b^{2 \log_b a}} \right) + \cdots + \left(\frac{a^L}{b^{L \log_b a}} \right) \right]$$

$$= c \cdot n^{\log_b a} \left[\overbrace{1 + 1 + 1 + \cdots + 1}^{L+1 \text{ terms}} \right]$$

$L = \log_b n$

$$\leq c \cdot n^{\log_b a} \cdot \log_b n = O(n^{\log_b a} \cdot \log n)$$

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

case 2: $c' n^{\log_b a} < f(n) < c n^{\log_b a}$

$$T(n) < cn^{\log_b a}\left[1 + \left(\frac{a}{b^{\log_b a}}\right) + \left(\frac{a^2}{b^{2\log_b a}}\right) + \cdots + \left(\frac{a^L}{b^{L\log_b a}}\right)\right]$$

$$= cn^{\log_b a}\left[1 + 1 + \cdots 1\right]$$

$$= cn^{\log_b a}\left[\log_b n\right] = O(n^{\log_b a}\log n)$$

Similar argument for lower bound.

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

case 3: $f(n) > dn^{\log_b a+\epsilon}$ And $\exists c$, $af(n/b) < cf(n)$

$c < 1$

$$af\left(\frac{n}{b}\right) < cf(n)$$

$$a^2 \cdot f\left(\frac{n}{b^2}\right) = a\left[a \cdot f\left(\frac{n}{b^2}\right)\right] < a\left[c \cdot f\left(\frac{n}{b}\right)\right] = c \cdot \left[a \cdot f\left(\frac{n}{b}\right)\right] < c \cdot c \cdot f(n)$$

by regularity applied to $f\left(\frac{n}{b^2}\right)$

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

case 3: $f(n) > dn^{\log_b a + \epsilon}$ And $\exists c,\ af(n/b) < cf(n)$

$$af\left(\frac{n}{b}\right) < cf(n)$$

$$a^2 f\left(\frac{n}{b^2}\right) = a\left[af\left(\frac{n}{b^2}\right)\right] < a\left[cf\left(\frac{n}{b}\right)\right] = c\left[af\left(\frac{n}{b}\right)\right] < c^2 f(n)$$

$$\underbrace{a^3 f\left(\frac{n}{b^3}\right)} < c \cdot a^2 f\left(\frac{n}{b^2}\right) < \underline{c^3 f(n)}$$

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

case 3: $f(n) > dn^{\log_b a + \epsilon}$ And $\exists c,\ af(n/b) < cf(n)$

$$T(n) < f(n) + c \cdot f(n) + c^2 \cdot f(n) + \cdots + c^L \cdot f(n)$$

$$= f(n)\left[\underbrace{1 + c + c^2 + \cdots + c^L}\right]$$

because $c \le 1$, this becomes a constant

$$= O(f(n))$$

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

case 3: $f(n) > dn^{\log_b a + \epsilon}$ And $\exists c,\ af(n/b) < cf(n)$

$$T(n) < f(n) + cf(n) + c^2 f(n) + \cdots + c^L f(n)$$

$$= f(n)\left[1 + c + c^2 + \cdots c^L\right]$$

$$= O(f(n))$$

It is important that c<1 for the sum term to be bounded by a constant

Similar argument for lower bound.

example from last class: $T(n) = \underbrace{\overset{16}{8}}_{a}T(n/\underbrace{2}_{b}) + \underbrace{\Theta(n^{2.99999})}_{f(n)}$

① compare $f(n)$ to $n^{\log_b a \pm \epsilon}$

$n^{2.999999}$ $\quad$ $n^{\log_b a} = n^{\log_2 8} = n^3$

because $2 < 3$, then $f(n) = O(n^{3-\epsilon})$

$\Rightarrow$ case 1 applies $\Rightarrow$ $T(n) = \Theta(n^3)$

example from last class: $T(n) = 8T(n/2) + \Theta(n^2)$

$$a = 8, b = 2, f(n) = \Theta(n^2)$$

example from last class: $T(n) = 8T(n/2) + \Theta(n^2)$

$$a = 8, b = 2, f(n) = \Theta(n^2)$$

Since $f(n) < cn^2 = O(n^{\log_2 8} - 0.1) = O(n^{2.9})$ then Case 1 applies.

Therefore $T(n) = \Theta(n^{\log_2 8}) = \Theta(n^3)$

1	7	8	9	★	1	4	3	2
a		b			c		d	

Schoolbook approach

$$T(n) = 4T(n/2) + 3O(n)$$

a b f(n)

$a =$

$f(n) = n^{\boxed{\ }}$

$n^{\log_2 4} = n^{\boxed{2}}$

Case I applies $\Rightarrow T(n) = \Theta(n^2)$

1 7 8 9 ★ 1 4 3 2

a b c d

Schoolbook approach

$$T(n) = 4T(n/2) + 3O(n)$$

$$a = 4, b = 2, f(n) = O(n)$$

Therefore, case1,

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

example:

$$T(n) = T\left(\frac{14}{17}n\right) + 24$$

$\frac{999}{1000}$

$f(n) = O(1)$

$a = 1 \qquad b = \frac{17}{14}$

compare

$$f(n) = O(1) \xleftrightarrow{\text{equal}} n^{\log_{17/14} 1} = n^0 = O(1).$$

$$\Rightarrow \text{case } 2 \Rightarrow T(n) = \Theta\left(n^{\log_{17/14} 1} \cdot \log n\right)$$

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

example:

$$T(n) = T\left(\frac{14}{17}n\right) + 24$$

Since $24 = \Theta(n^{\log_{17/14} 1}) = \Theta(n^0)$, case 2 applies.

example:

$$T(n) = T\left(\frac{14}{17}n\right) + 24$$

Since $24 = \Theta(n^{\log_{17/14} 1}) = \Theta(n^0)$, case 2 applies.

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

$$T(n) = \underbrace{\boxed{2}}_{a}T(n/\underbrace{\boxed{2}}_{b}) + \underbrace{n^3}_{f(n)}$$

compare

$$\underline{n^3} \quad \text{to} \quad n^{\log_2 2} = \underline{n^1}$$

$$a \cdot f\left(\frac{n}{b}\right) = 2 \cdot \left(\frac{n}{2}\right)^3 = 2\left(\frac{n^3}{8}\right) = \frac{n^3}{4} < \overset{??}{c} \cdot n^3$$

So regularity condition holds

Set $c = \frac{1}{2} < 1$.

$$\Rightarrow \quad T(n) = \Theta(n^3)$$

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

Since $n^3 = \Omega(n^{\log_2 2+\epsilon})$ and $2\left(\frac{n}{2}\right)^3 < \left(\frac{1}{2}\right) n^3$ Case 3 applies.

$$T(n) = \underbrace{16}_{a}T(n/\underbrace{4}_{b}) + \underbrace{n^2}_{f(n)}$$

$$f(n) = n^2 \xleftrightarrow{\text{equal}} n^{\log_b a} = n^{\log_4 16} = n^2$$

$$\underline{\text{case } 2} \Rightarrow \quad T(n) = \Theta(n^2 \cdot \log n)$$

$$T(n) = 7T(n/2) + \Theta(n^2)$$

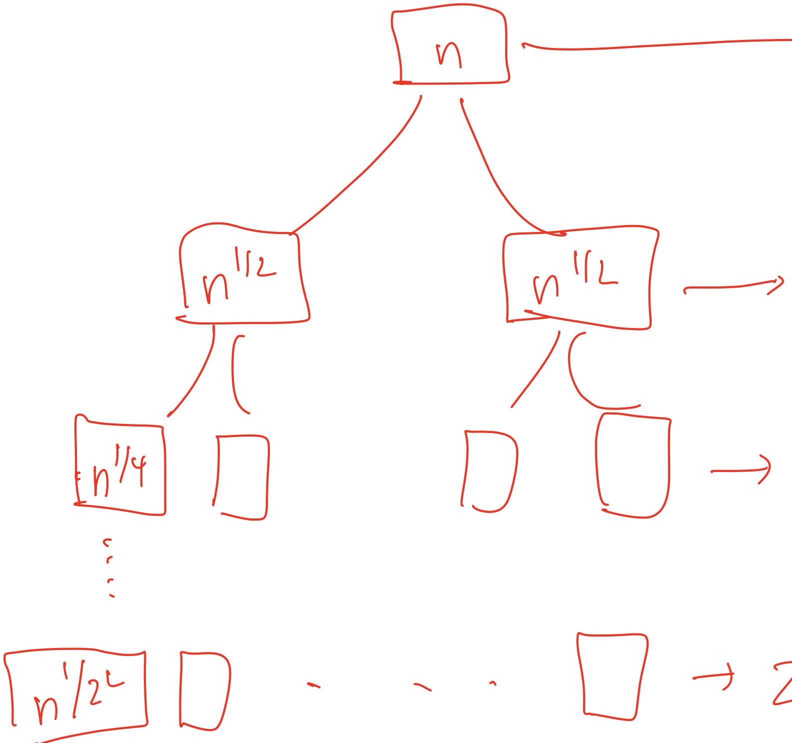
for practise.

$\Rightarrow$ corresponds to a problem we will encounter next class.

Change of
Variable

$\sqrt{n} = n^{1/2}$

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



$\lg n$

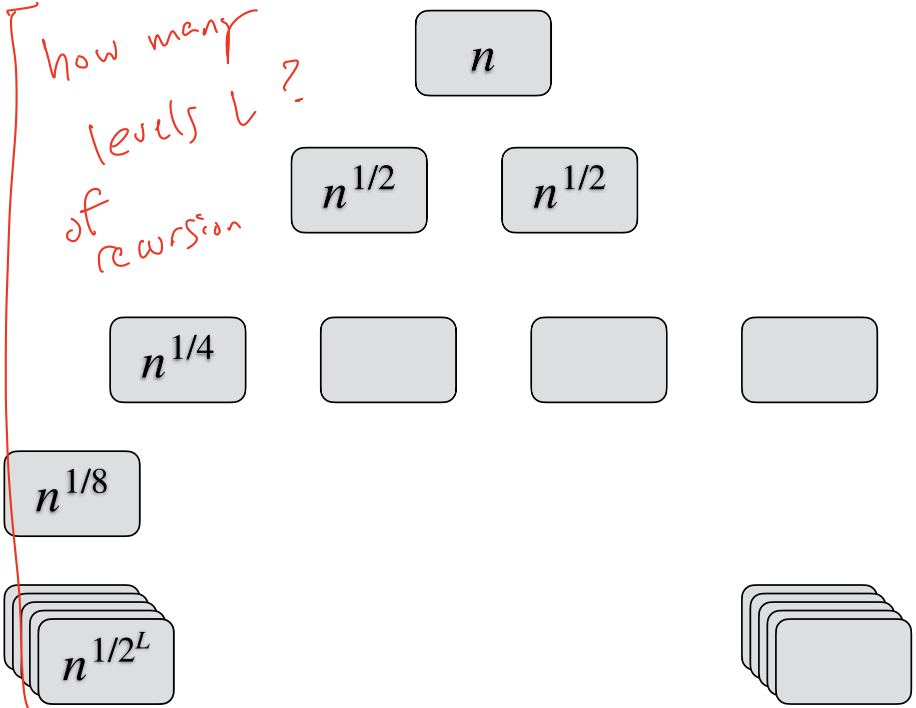
$2 \cdot \lg(n^{1/2}) = \cancel{2}\left(\cancel{\tfrac{1}{2}}\right) \cdot \lg(n)$

$4 \cdot \lg(n^{1/4}) = \cancel{4}\left(\cancel{\tfrac{1}{4}}\right) \cdot \lg(n)$

$2^c \cdot \lg\left(n^{1/2^c}\right) = \cancel{2^c \tfrac{1}{2^c}} \cdot \lg(n)$

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

how many levels L? of recursion



$\lg n$ $= \lg n$

$2\lg(n^{1/2})$ $= \lg n$

$2^2\lg(n^{1/2^2})$

$2^3\lg(n^{1/2^3})$ $\vdots$

$2^L\lg(n^{1/2^L})$ $= \lg(n)$

How to solve for L?

$$\log\left(n^{\frac{1}{2^L}}\right) = 2$$

Take logs on both sides:

$$\left(\frac{1}{2^L}\right)\log(n) = \log(2) = 1$$

$$\Rightarrow \log(n) = 2^L$$

$$\Rightarrow \log(\log(n)) = L$$

$$L = \log\log(n)$$

How to solve for L?

$$n^{\frac{1}{2^L}} = 2$$

Take logs on both sides:

$$\frac{1}{2^L} \log n = \log(2)$$

Then multiply both sides by 2^L, and take logs again.

$$\log \log n = L$$

How to solve for L?

$$n^{\frac{1}{2^L}} = 2$$

For our purposes, this value can be a constant. Why not 1?

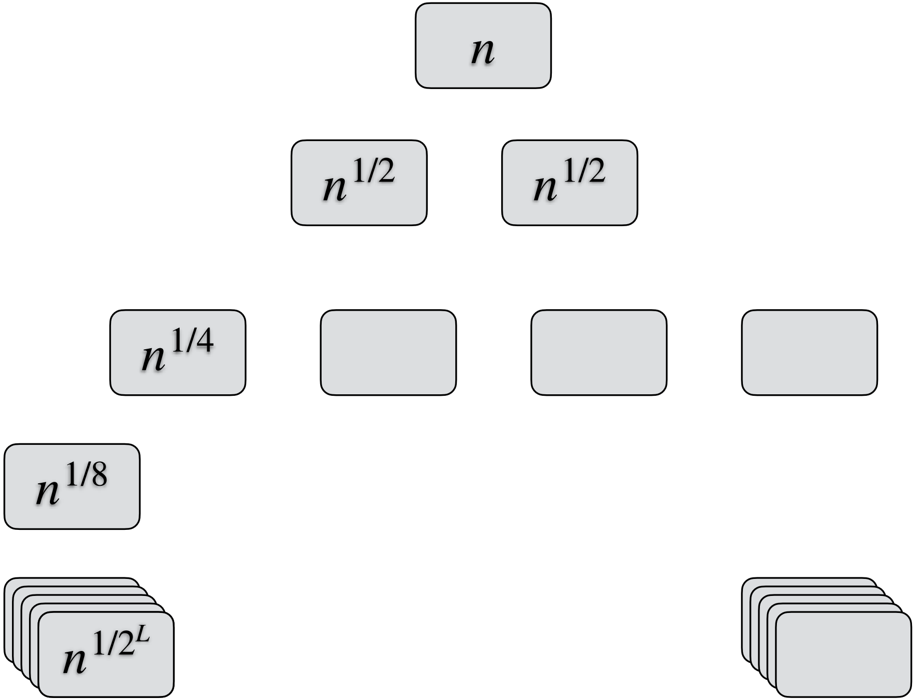
Take logs on both sides:

$$\frac{1}{2^L} \log n = \log(2)$$

Then multiply both sides by 2^L, and take logs again.

$$\log \log n = L$$

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



$\lg n$

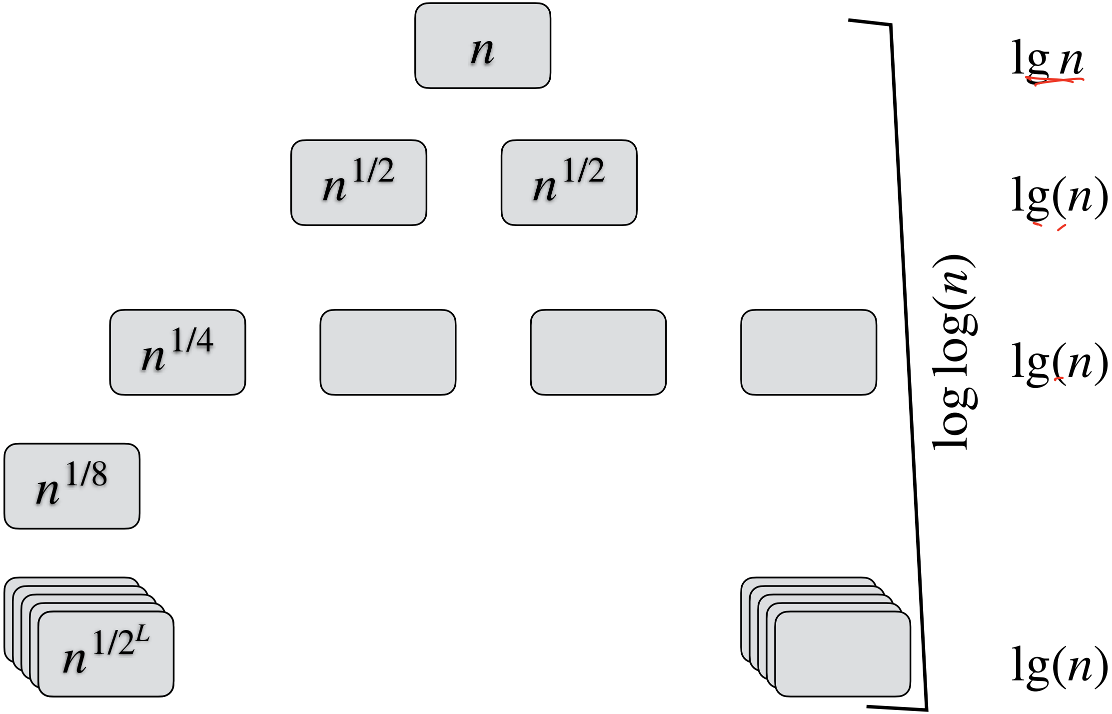
$2\lg(n^{1/2})$

$2^2\lg(n^{1/2^2})$

$2^3\lg(n^{1/2^3})$

$2^L\lg(n^{1/2^L})$

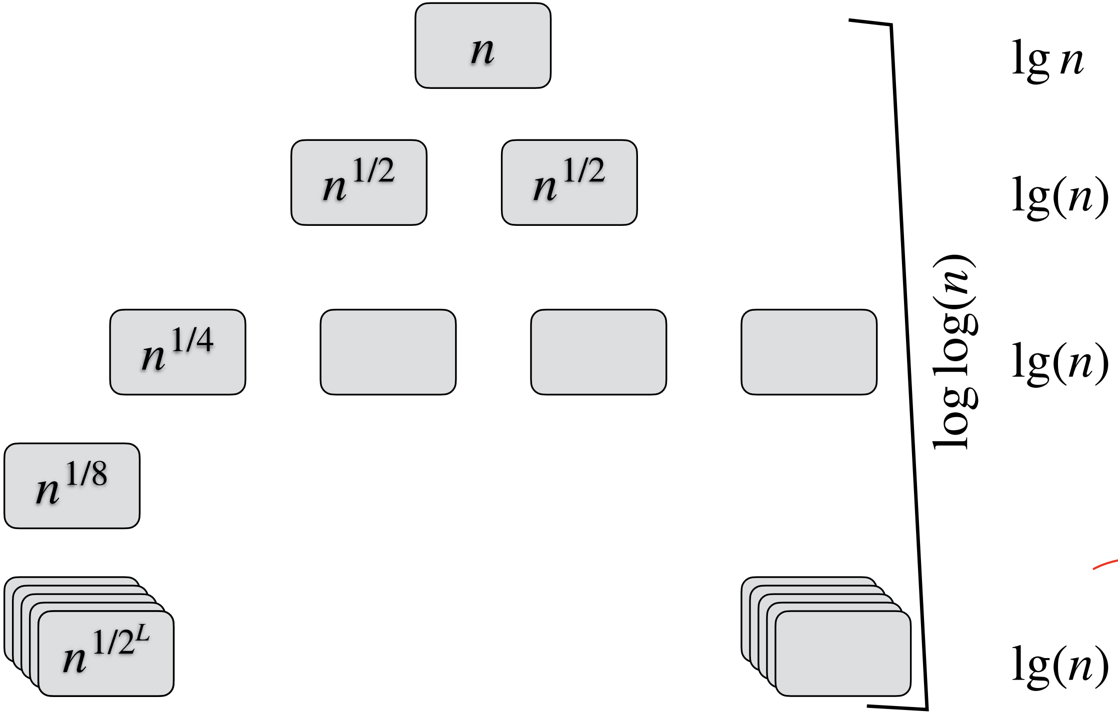
$$T(n) = 2T\left(\sqrt{n}\right) + \lg n$$



$T(n) =$

$\Theta(\log n \cdot \log\log n)$

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



Sums to $O(\log n \log\log n)$

~~Use induction to~~ prove.

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

Lets rewrite with $m = \log n$

$T(2^m) = T(n)$

$T(2^m) = 2T(2^{m/2}) + \cancel{c} \cdot m$

Define $S(m) = T(2^m)$, $S(m/2) = T(2^{m/2})$ New recurrence.

$S(m) = 2 \cdot S(m/2) + m$

By case 2, $S(m) = \Theta(m \cdot \log m)$

Finally, since $m = \log(n)$

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

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

Lets rewrite with $m = \log n$

$$T(2^m) = 2T(2^{m/2}) + c \cdot m$$

Define $S(m) = T(2^m)$

$$S(m) = 2S(m/2) + \Theta(m)$$

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

Lets rewrite with $m = \log n$

$$T(2^m) = 2T(2^{m/2}) + c \cdot m$$

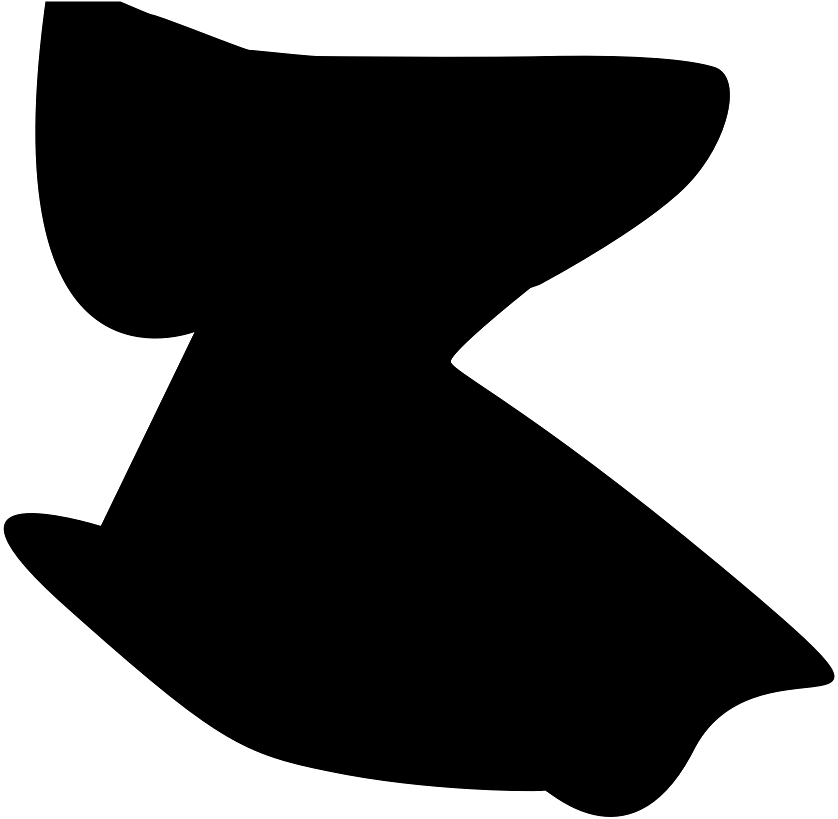
Define $S(m) = T(2^m)$

$$S(m) = 2S(m/2) + \Theta(m)$$

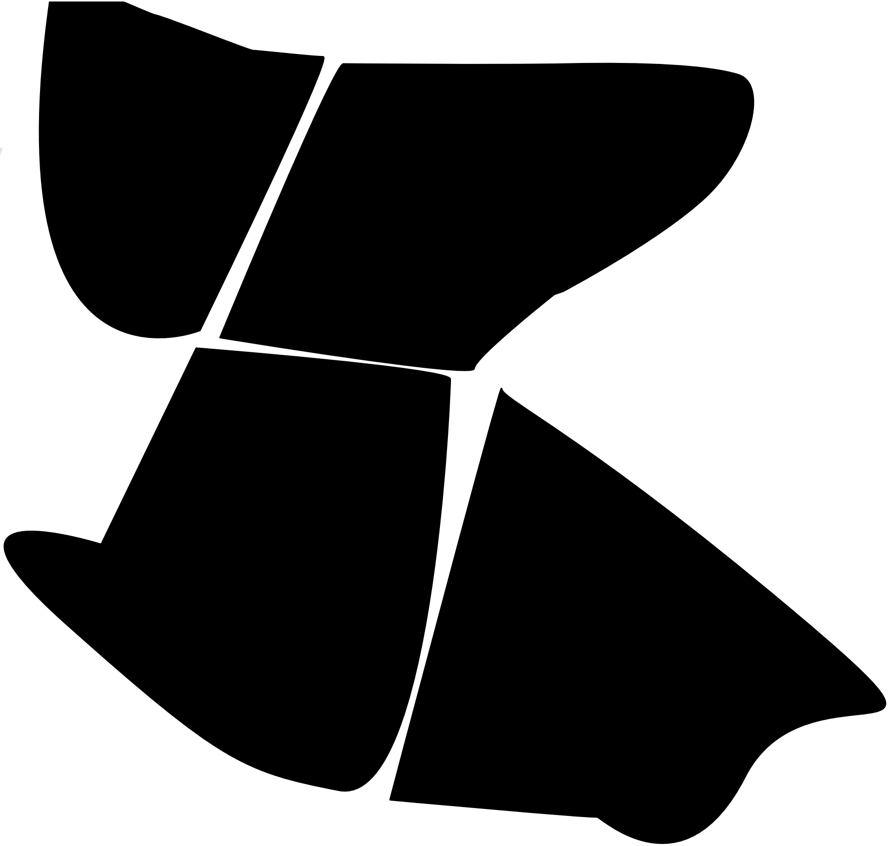
Apply Master's Thm case 2: $S(m) = \Theta(m \log m)$

Since $m = \log n$, we have $T(n) = \Theta(\log n \log \log n)$

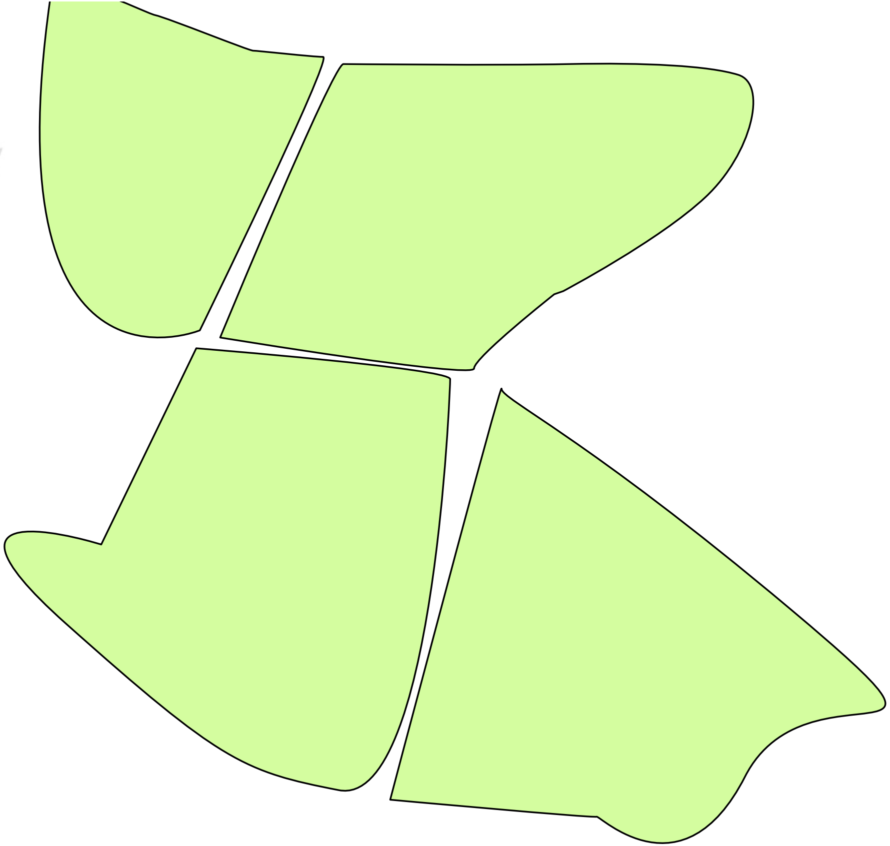
divide
& conquer



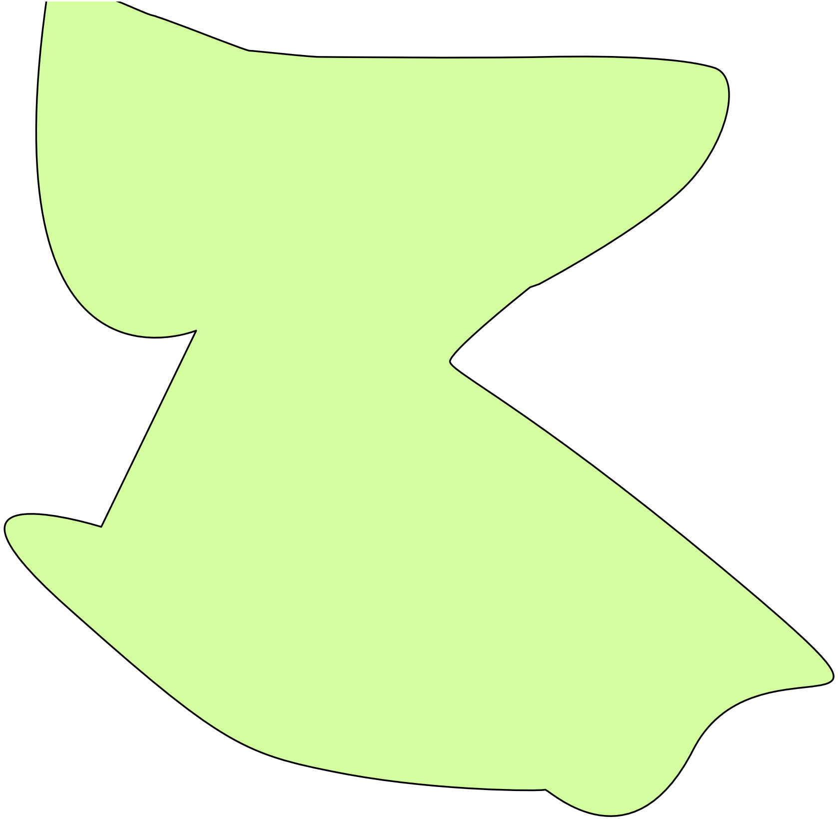
http://www.kitchenknifedrawer.com/files/1696205/uploaded/K6615D.jpg



http://www.kitchenknifedrawer.com/files/1696205/uploaded/K6615D.jpg



http://www.kitchenknifedrawer.com/files/1696205/uploaded/K6615D.jpg



http://www.kitchenknifedrawer.com/files/1696205/uploaded/K6615D.jpg

Examples we will discuss

- Mergesort
- Arbitrage
- Closest pair of points
- Matrix mult
- Fast Fourier Transform

Merge

merge-sort (A, p, r)
if $p < r$

$q \leftarrow \lfloor (p+r)/2 \rfloor$

merge-sort (A, p, q)
merge-sort $(A, q+1, r)$
merge(A, p, q, r)

MERGE$(A[1..n], m)$:
$i \leftarrow 1;\ j \leftarrow m+1$
for $k \leftarrow 1$ to n
if $j > n$
$B[k] \leftarrow A[i];\ i \leftarrow i+1$
else if $i > m$
$B[k] \leftarrow A[j];\ j \leftarrow j+1$
else if $A[i] < A[j]$
$B[k] \leftarrow A[i];\ i \leftarrow i+1$
else
$B[k] \leftarrow A[j];\ j \leftarrow j+1$

for $k \leftarrow 1$ to n
$A[k] \leftarrow B[k]$

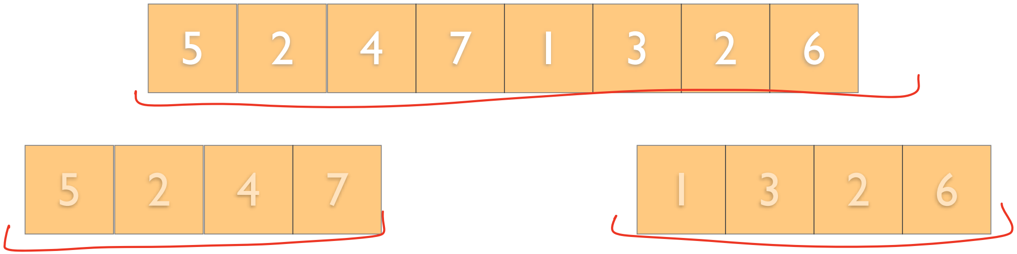
jeff erickson

5	2	4	7	1	3	2	6

merge-sort (A, p, r)
  if $p < r$
    $q \leftarrow \lfloor (p + r)/2 \rfloor$
    merge-sort (A, p, q)
    merge-sort $(A, q + 1, r)$
    merge(A, p, q, r)

```
MERGE(A[1..n], m):
  i ← 1; j ← m + 1
  for k ← 1 to n
      if j > n
          B[k] ← A[i]; i ← i + 1
      else if i > m
          B[k] ← A[j]; j ← j + 1
      else if A[i] < A[j]
          B[k] ← A[i]; i ← i + 1
      else
          B[k] ← A[j]; j ← j + 1
  for k ← 1 to n
      A[k] ← B[k]
```

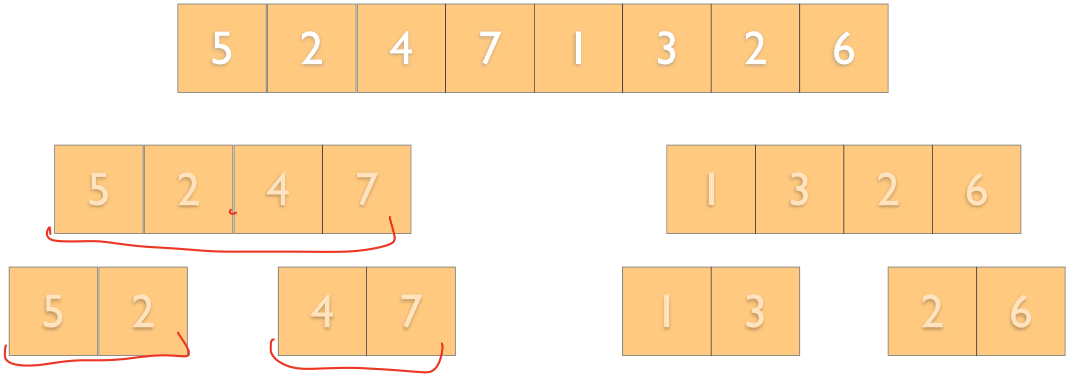
jeff erickson



merge-sort (A, p, r)
    if $p < r$
        $q \leftarrow \lfloor (p+r)/2 \rfloor$
        merge-sort (A, p, q)
        merge-sort $(A, q+1, r)$
        merge(A, p, q, r)

MERGE$(A[1..n], m)$:
  $i \leftarrow 1$; $j \leftarrow m+1$
  for $k \leftarrow 1$ to n
    if $j > n$
      $B[k] \leftarrow A[i]$; $i \leftarrow i+1$
    else if $i > m$
      $B[k] \leftarrow A[j]$; $j \leftarrow j+1$
    else if $A[i] < A[j]$
      $B[k] \leftarrow A[i]$; $i \leftarrow i+1$
    else
      $B[k] \leftarrow A[j]$; $j \leftarrow j+1$
  for $k \leftarrow 1$ to n
    $A[k] \leftarrow B[k]$

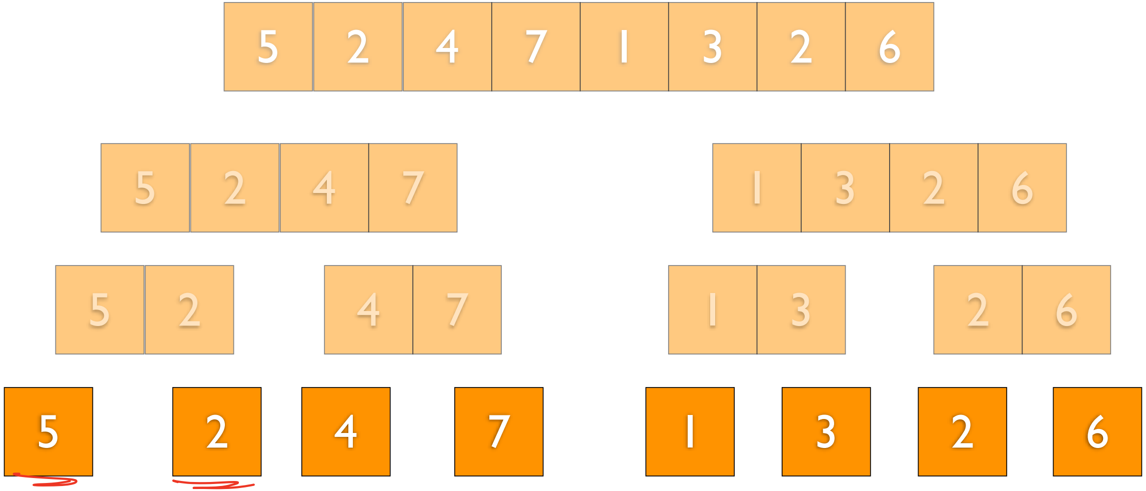
jeff erickson



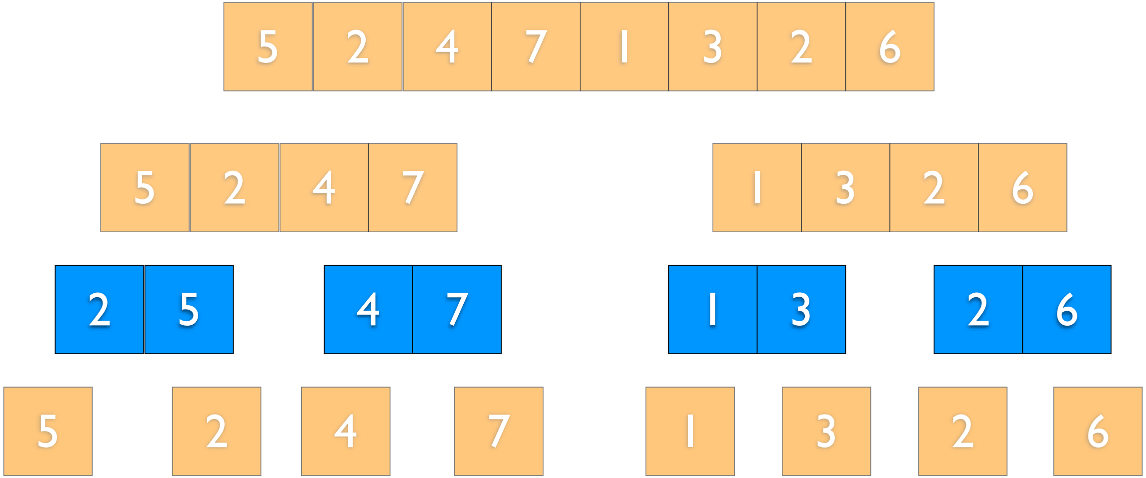
merge-sort (A, p, r)
  if $p < r$
    $q \leftarrow \lfloor (p+r)/2 \rfloor$
    merge-sort (A, p, q)
    merge-sort $(A, q+1, r)$
    merge(A, p, q, r)

MERGE$(A[1..n], m)$:
  $i \leftarrow 1$; $j \leftarrow m+1$
  for $k \leftarrow 1$ to n
    if $j > n$
      $B[k] \leftarrow A[i]$; $i \leftarrow i+1$
    else if $i > m$
      $B[k] \leftarrow A[j]$; $j \leftarrow j+1$
    else if $A[i] < A[j]$
      $B[k] \leftarrow A[i]$; $i \leftarrow i+1$
    else
      $B[k] \leftarrow A[j]$; $j \leftarrow j+1$
  for $k \leftarrow 1$ to n
    $A[k] \leftarrow B[k]$

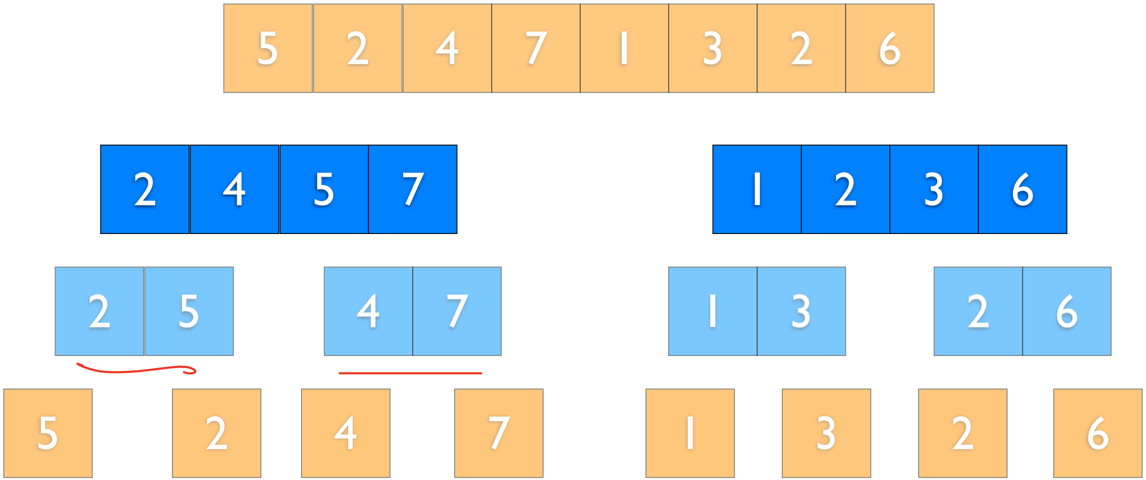
jeff erickson



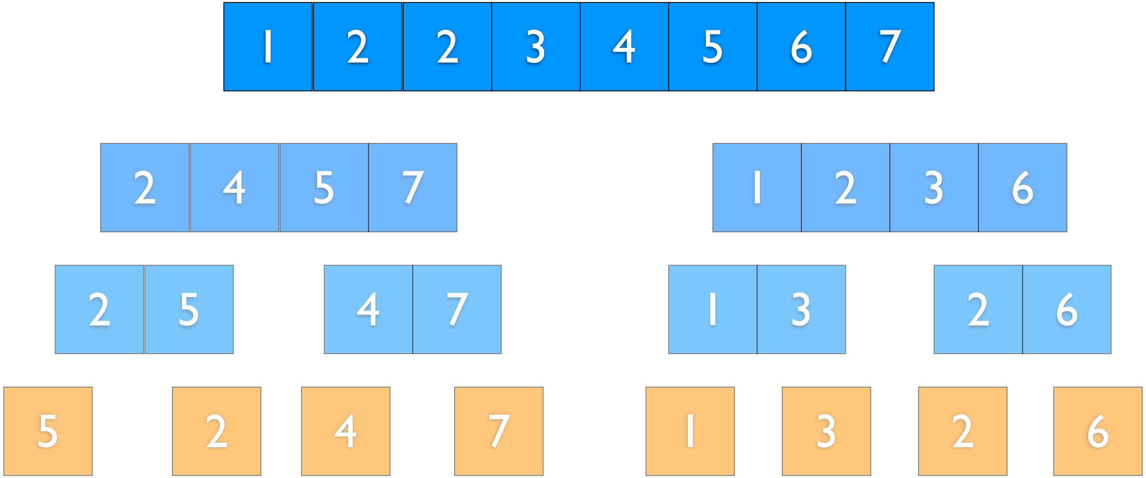
merge-sort (A, p, r)
    if $p < r$
        $q \leftarrow \lfloor (p+r)/2 \rfloor$
        merge-sort (A, p, q)
        merge-sort $(A, q+1, r)$
        merge(A, p, q, r)



merge-sort (A, p, r)
  if $p < r$
    $q \leftarrow \lfloor (p + r)/2 \rfloor$
    merge-sort (A, p, q)
    merge-sort $(A, q + 1, r)$
    merge(A, p, q, r)



merge-sort (A, p, r)
  if $p < r$
    $q \leftarrow \lfloor (p+r)/2 \rfloor$
    merge-sort (A, p, q)
    merge-sort $(A, q+1, r)$
    merge(A, p, q, r)



merge-sort (A, p, r)
  if $p < r$
    $q \leftarrow \lfloor (p+r)/2 \rfloor$
    merge-sort (A, p, q)
    merge-sort $(A, q+1, r)$
    merge(A, p, q, r)

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

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