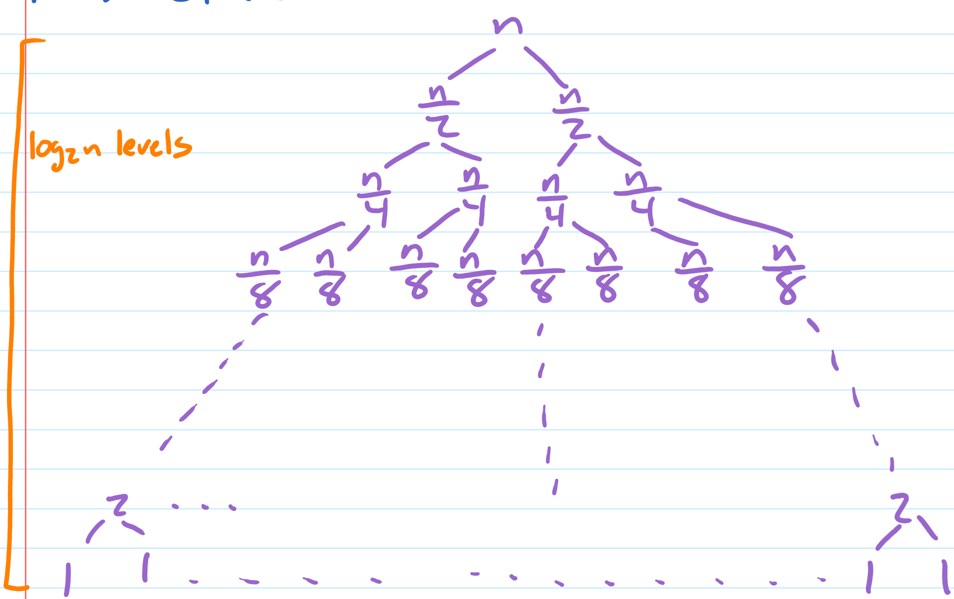
Mergesort Recursion Tree

$T(n)$ = worst-case # of key comparisons to sort n items

$T(1) = 0$

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



subproblems

size	number	work
n	1	$n-1$
$\frac{n}{2}$	2	$2\cdot\left(\frac{n}{2}-1\right)=n-2$
$\frac{n}{4}$	4	$4\cdot\left(\frac{n}{4}-1\right)=n-4$
$\frac{n}{8}$	8	$8\cdot\left(\frac{n}{8}-1\right)=n-8$
	$\vdots$	$\vdots$
2	$\frac{n}{2}$	$\frac{n}{2}(2-1)=n-\frac{n}{2}$
1	$2^{\log_2 n}=n$	$\underline{n\cdot 0=0}$

```
mergesort(A)
   sort first half
   sort last half
   merge sorted halves and return
```

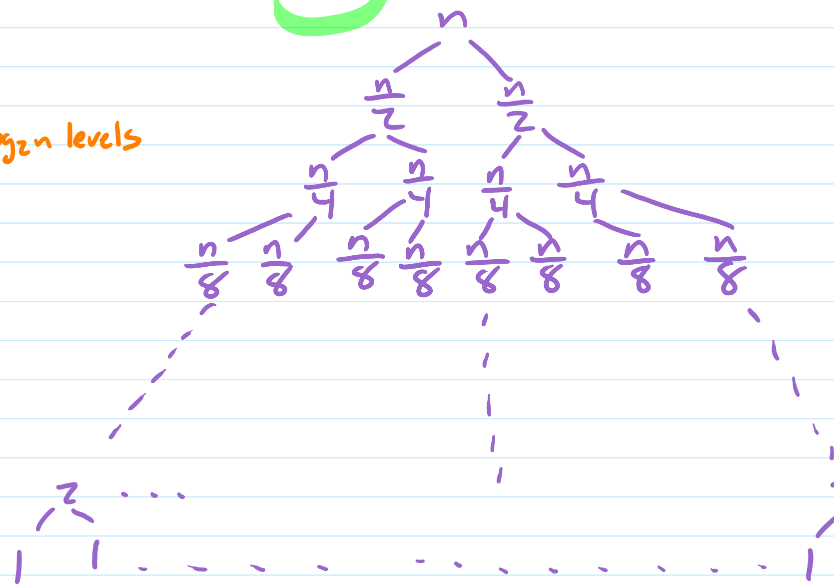
Mergesort Recursion Tree

$T(n)$ = worst-case # of key comparisons to sort n items

$T(1) = 0$

$T(n) = 2T(\frac{n}{2}) + n-1$ medium

$\log_2 n$ levels



subproblems size	number	work
n	1	$n-1$
$\frac{n}{2}$	2	$2\cdot(\frac{n}{2}-1)=n-2$
$\frac{n}{4}$	4	$4\cdot(\frac{n}{4}-1)=n-4$
$\frac{n}{8}$	8	$8\cdot(\frac{n}{8}-1)=n-8$
$\vdots$	$\vdots$	$\vdots$
2	$\frac{n}{2}$	$\frac{n}{2}(2-1)=n-\frac{n}{2}$
1	$2^{\log_2 n}=n$	$n\cdot 0=0$

for n a power of 2, $T(n) = n\log_2 n - (n-1)$

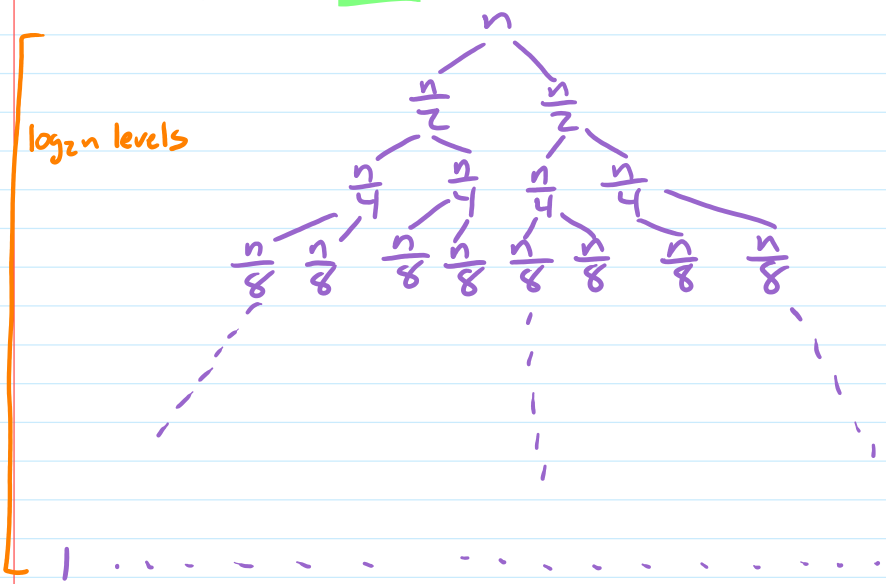
in general, $T(n) \in \Theta(n \log n)$

```
mergesort(A)
   sort first half
   sort last half
   merge sorted halves and return
```

Market Timing Recursion Tree

$T(1) = c_1$

$T(n) = 2T(\frac{n}{2}) + c_2$



subproblems

size	number	work
n	1	$1 \cdot c_2$
$\frac{n}{2}$	2	$2 \cdot c_2$
$\frac{n}{4}$	4	$4 \cdot c_2$
$\frac{n}{8}$	8	$8 \cdot c_2$
⋮	⋮	⋮
1	$2^{\log_2 n} = n$	$n \cdot c_1$

increasing work

$T(n) = c_1 \cdot n + c_2 \cdot (n-1)$

$T(n) \in \Theta(n)$

```
opt_profit(A)
   minL, maxL, profitL = opt_profit(1st half)
   minR, maxR, profitR = opt_profit(2nd half)
   return min(minL, minR), max(maxL, maxR), max(profitL, profitR, maxR - minL)
```

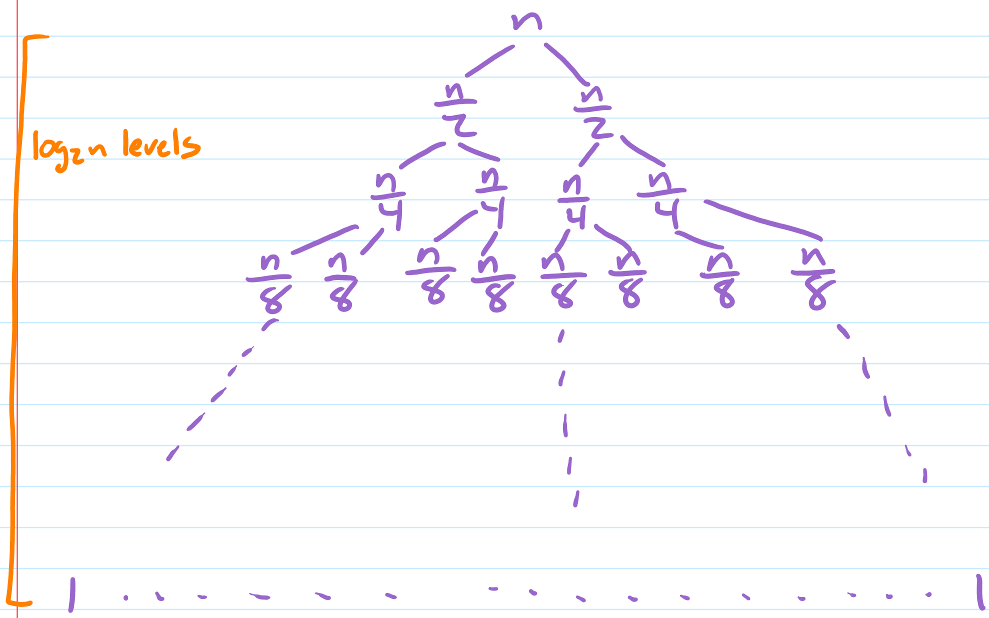
specify halves by range of indices

$O(1)$

Market Timing Recursion Tree

$T(1) = c_1$

$T(n) = 2T\left(\frac{n}{2}\right) + c_2$



subproblems

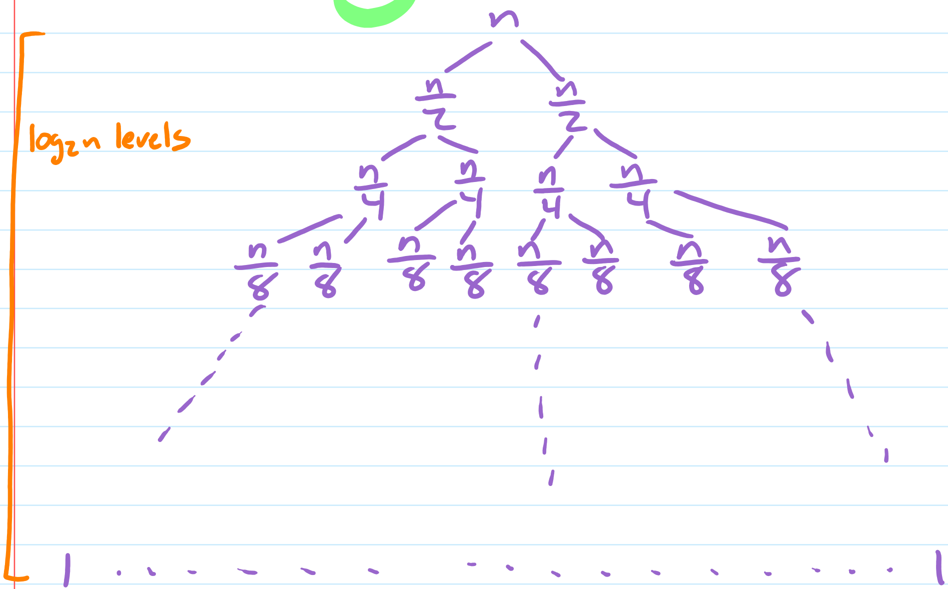
size	number	work
n	1	$1 \cdot c_2$
$\frac{n}{2}$	2	$2 \cdot c_2$
$\frac{n}{4}$	4	$4 \cdot c_2$
$\frac{n}{8}$	8	$8 \cdot c_2$
$\vdots$	$\vdots$	$\vdots$
1	$2^{\log_2 n} = n$	$\underline{n \cdot c_1}$

```
opt_profit(A)
   minL, maxL, profitL = opt_profit(1st half)
   minR, maxR, profitR = opt_profit(2nd half)
   return min(minL, minR), max(maxL, maxR), max(profitL, profitR, maxR - minL)
```

Market Timing Recursion Tree

$T(1) = c_1$

$T(n) = 2T(\frac{n}{2}) + c_2$ easy



subproblems

size	number	work
n	1	$1 \cdot c_2$
$\frac{n}{2}$	2	$2 \cdot c_2$
$\frac{n}{4}$	4	$4 \cdot c_2$
$\frac{n}{8}$	8	$8 \cdot c_2$
1	$2^{\log_2 n} = n$	$n \cdot c_1$

$T(n) = c_1 \cdot n + c_2 \cdot (n-1)$

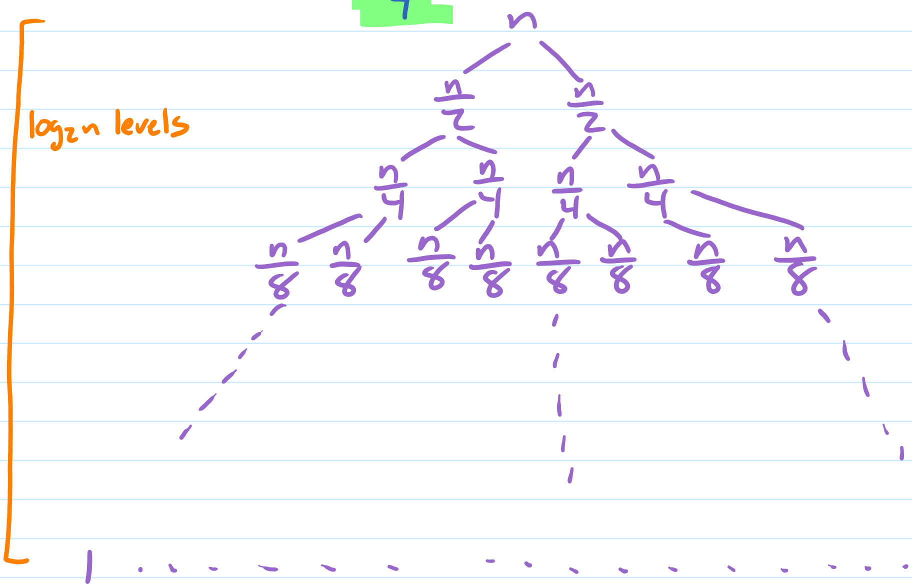
$T(n) \in \Theta(n)$

```
opt_profit(A)
   minL, maxL, profitL = opt_profit(1st half)
   minR, maxR, profitR = opt_profit(2nd half)
   return min(minL, minR), max(maxL, maxR), max(profitL, profitR, maxR - minL)
```

Naïve Inversion Counting Recursion Tree

$T(1) = 0$

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



decreasing faster

increasing

subproblems size	subproblems number	work
n	1	$\frac{n^2}{4}$
$\frac{n}{2}$	2	$2 \cdot \frac{1}{4}\left(\frac{n}{2}\right)^2 = \frac{n^2}{8}$
$\frac{n}{4}$	4	$4 \cdot \frac{1}{4}\left(\frac{n}{4}\right)^2 = \frac{n^2}{16}$
$\frac{n}{8}$	8	$8 \cdot \frac{1}{4}\left(\frac{n}{8}\right)^2 = \frac{n^2}{32}$
$\vdots$	$\vdots$	$\vdots$
1	$2^{\log_2 n} = n$	$n \cdot \frac{1}{4}(1)^2 = \frac{n}{4}$

$\frac{n^2}{4} \le T(n) \le \frac{n^2}{2}$

$T(n) \in \Theta(n^2)$

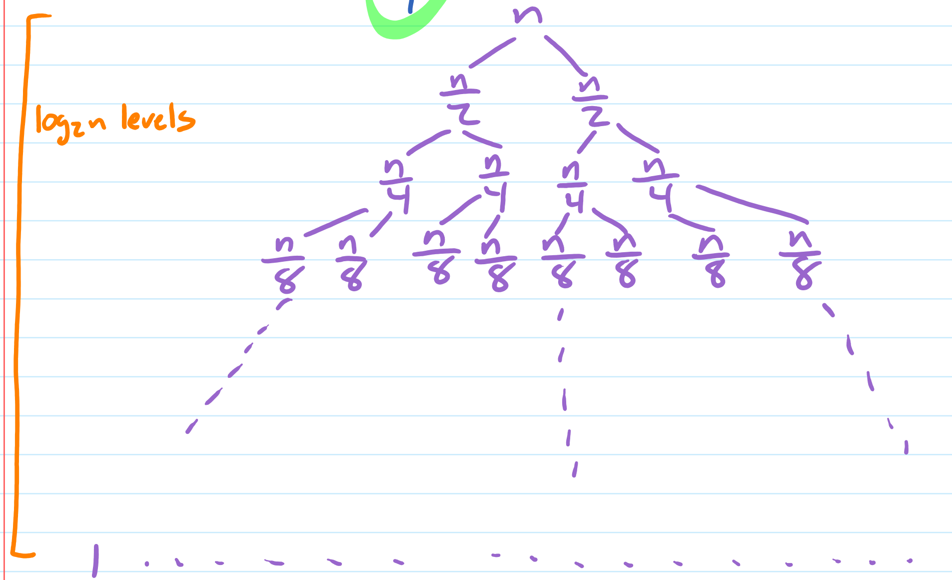
```
count_inversions(A)
   countL = count_inversions(1st half)
   countR = count_inversions(2nd half)
   countAcross = number of inversions with 1 elt in 1st half other in 2nd half
   return countL + countR + countAcross
```

Naïve Inversion Counting Recursion Tree

hard

$T(1) = 0$

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



subproblems

size	number	work
n	1	$\frac{n^2}{4}$
$\frac{n}{2}$	2	$2 \cdot \frac{1}{4}\left(\frac{n}{2}\right)^2 = \frac{n^2}{8}$
$\frac{n}{4}$	4	$4 \cdot \frac{1}{4}\left(\frac{n}{4}\right)^2 = \frac{n^2}{16}$
$\frac{n}{8}$	8	$8 \cdot \frac{1}{4}\left(\frac{n}{8}\right)^2 = \frac{n^2}{32}$
$\vdots$	$\vdots$	$\vdots$
1	$2^{\log_2 n} = n$	$n \cdot \frac{1}{4}(1)^2 = \frac{n}{4}$

$\frac{n^2}{4} \leq T(n) \leq \frac{n^2}{2}$

$T(n) \in \Theta(n^2)$

```
count_inversions(A)
   countL = count_inversions(1st half)
   countR = count_inversions(2nd half)
   countAcross = number of inversions with 1 elt in 1st half other in 2nd half
   return countL + countR + countAcross
```

Generic Recursion Tree

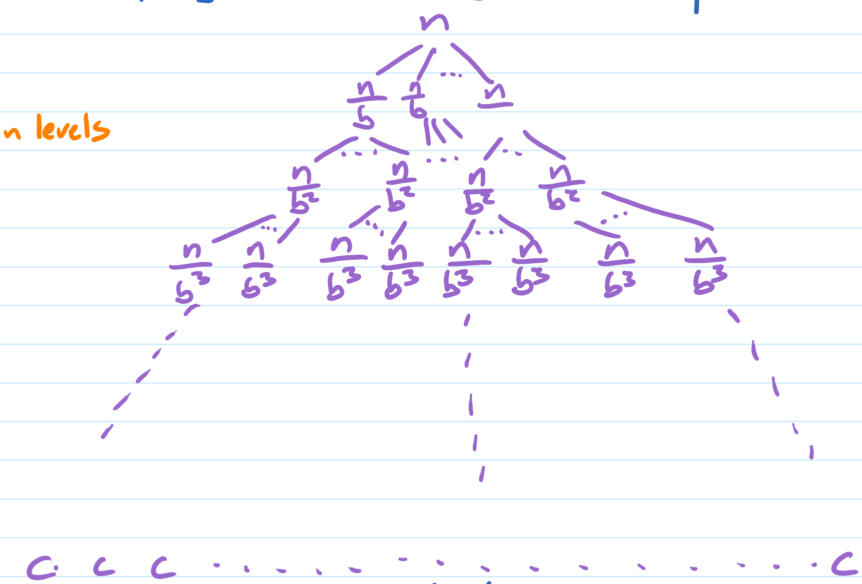
$T(1) = c$

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

a = # of subproblems

$\frac{n}{b}$ = size of subproblems

$\log_b n$ levels



subproblems size	number	work
n	1	n^d
$\frac{n}{b}$	a	$\left(\frac{n}{b}\right)^d \cdot a = \left(\frac{a}{b^d}\right)^1 \cdot n^d$
$\frac{n}{b^2}$	a^2	$\left(\frac{n}{b^2}\right)^d \cdot a^2 = \left(\frac{a}{b^d}\right)^2 \cdot n^d$
$\frac{n}{b^3}$	a^3	$\left(\frac{n}{b^3}\right)^d \cdot a^3 = \left(\frac{a}{b^d}\right)^3 \cdot n^d$
$\vdots$	$\vdots$	$\vdots$
1	$a^{\log_b n} = n^{\log_b a}$	$c \cdot n^{\log_b a}$

$T(n)$ is $\Theta(n^d)$

$a = b^d$

$\log_b a = d$

$d > \log_b a$

$b^d > a$

$\frac{a}{b^d} < 1$

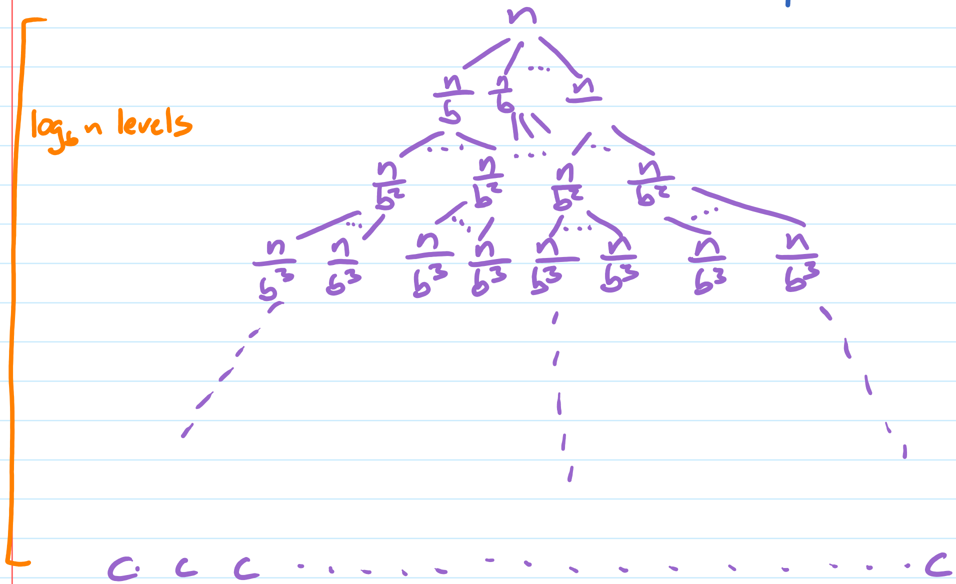
Generic Recursion Tree

$T(1) = c$

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

a = # of subproblems

$\frac{n}{b}$ = size of subproblems



subproblems size	subproblems number	work
n	1	n^d
$\frac{n}{b}$	a	$\left(\frac{n}{b}\right)^d \cdot a = \frac{a}{b^d} \cdot n^d$
$\frac{n}{b^2}$	a^2	$\left(\frac{n}{b^2}\right)^d \cdot a^2 = \left(\frac{a}{b^d}\right)^2 \cdot n^d$
$\frac{n}{b^3}$	a^3	$\left(\frac{n}{b^3}\right)^d \cdot a^3 = \left(\frac{a}{b^d}\right)^3 \cdot n^d$
$\vdots$	$\vdots$	$\vdots$
1	$a^{\log_b n} = n^{\log_b a}$	$\underline{c \cdot n^{\log_b a}}$

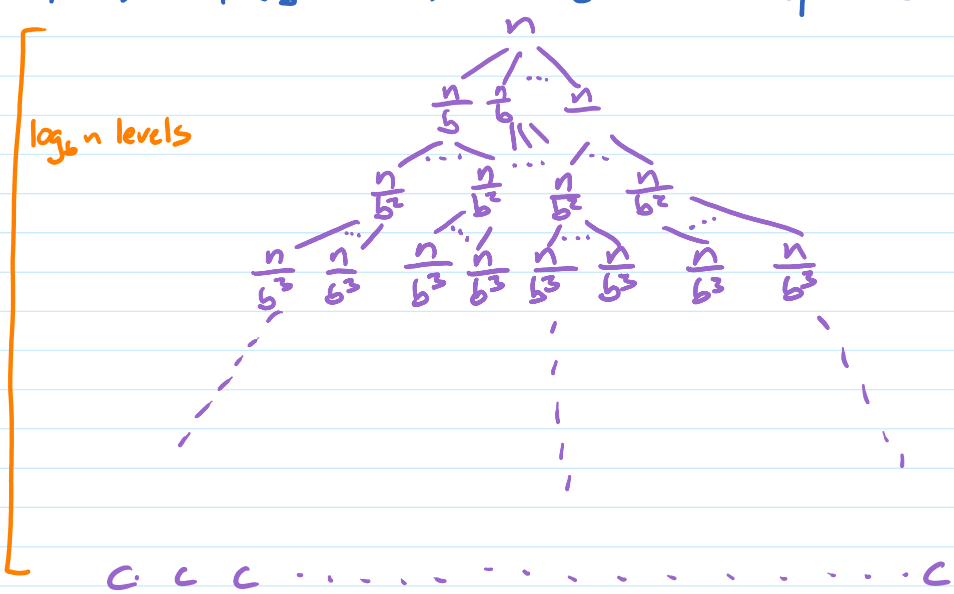
Generic Recursion Tree

$T(1) = c$

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

$a = \#$ of subproblems

$\frac{n}{b} =$ size of subproblems



subproblems size	subproblems number	work
n	1	$f(n)$
$\frac{n}{b}$	a	$a \cdot f\left(\frac{n}{b}\right)$
$\frac{n}{b^2}$	a^2	$a^2 \cdot f\left(\frac{n}{b^2}\right)$
$\frac{n}{b^3}$	a^3	$a^3 \cdot f\left(\frac{n}{b^3}\right)$
1	$a^{\log_b n} = n^{\log_b a}$	$c \cdot n^{\log_b a}$

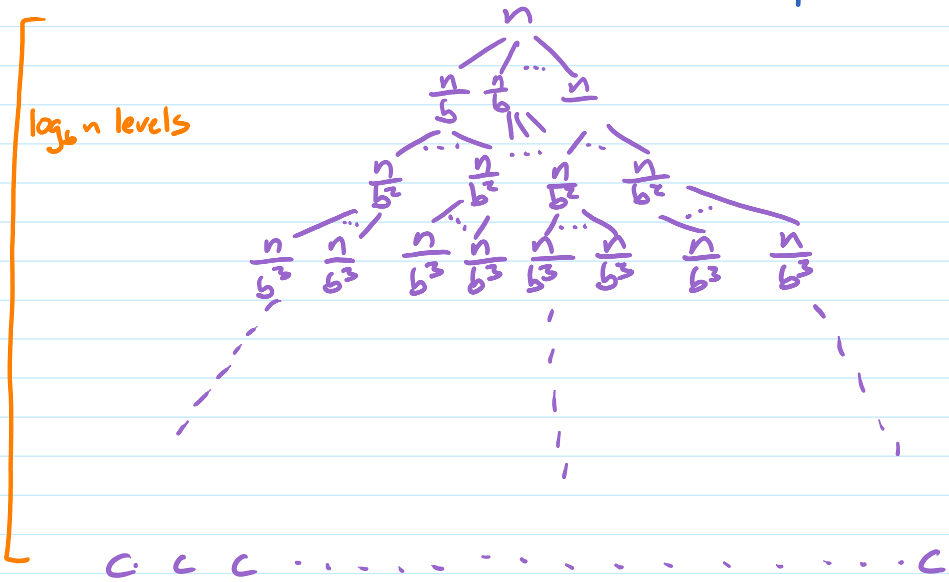
Generic Recursion Tree

$T(1) = c$

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

a = # of subproblems

$\frac{n}{b}$ = size of subproblems

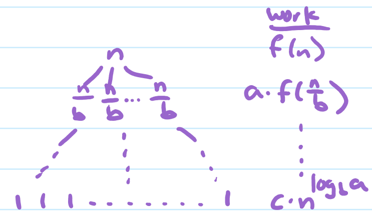


subproblems size	number	work
n	1	$f(n)$
$\frac{n}{b}$	a	$a \cdot f\left(\frac{n}{b}\right)$
$\frac{n}{b^2}$	a^2	$a^2 \cdot f\left(\frac{n}{b^2}\right)$
$\frac{n}{b^3}$	a^3	$a^3 \cdot f\left(\frac{n}{b^3}\right)$
⋮	⋮	⋮
1	$a^{\log_b n} = n^{\log_b a}$	$c \cdot n^{\log_b a}$

How does $f(n)$ relate to $n^{\log_b a}$?

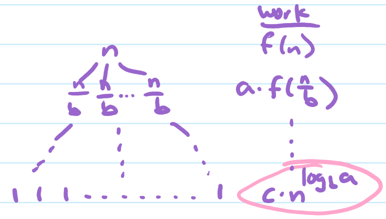
Master Theorem

Suppose $$T(n) = \underbrace{a}_{\#\text{ of subproblems}} \cdot T\Big(\overbrace{\tfrac{n}{b}}^{\text{size of subproblems}}\Big) + \underbrace{f(n)}_{\text{divide+combine work}}$$



Then

Master Theorem



Suppose $T(n) = \underbrace{a}_{\text{\# of subproblems}} \cdot T\Big(\overbrace{\tfrac{n}{b}}^{\text{size of subproblems}}\Big) + \underbrace{f(n)}_{\text{divide+combine work}}$

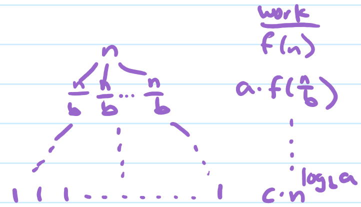
Then

if $f(n) \in O(n^{\log_b a - \varepsilon})$ for some $\varepsilon > 0$ then $T(n) \in \Theta(n^{\log_b a})$

base cases dominate

Master Theorem

Suppose $T(n) = \underbrace{a}_{\text{\# of subproblems}} \cdot T\big(\overbrace{\tfrac{n}{b}}^{\text{size of subproblems}}\big) + \underbrace{f(n)}_{\text{divide+combine work}}$



Then

if $f(n) \in O(n^{\log_b a - \varepsilon})$ for some $\varepsilon > 0$ then $T(n) \in \Theta(n^{\log_b a})$

base cases dominate

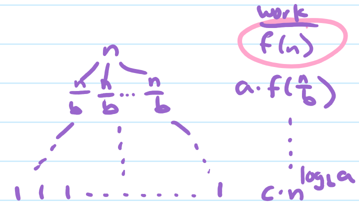
if $f(n) \in \Theta(n^{\log_b a})$ then $T(n) \in$

Master Theorem

size of subproblems

Suppose $T(n) = \underbrace{a}_{\#\text{ of subproblems}} \cdot T\left(\frac{n}{b}\right) + \underbrace{f(n)}_{\text{divide+combine work}}$ $\Theta(n^d)$

$\frac{n}{b}$ can be $\lfloor \frac{n}{b} \rfloor$, $\lceil \frac{n}{b} \rceil$, $\frac{n}{b} \pm c$



Then

$d < \log_b a$

if $f(n) \in O(n^{\log_b a - \varepsilon})$ for some $\varepsilon > 0$ then $T(n) \in \Theta(n^{\log_b a})$

base cases dominate

$d = \log_b a$

if $f(n) \in \Theta(n^{\log_b a})$ then $T(n) \in \Theta(n^{\log_b a} \cdot \log n)$

balanced

$d > \log_b a$

if $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ for some $\varepsilon > 0$ $T(n) \in \Theta(f(n))$

and if $a \cdot f(\frac{n}{b}) \le c \cdot f(n)$ for some $c > 1$ and all large n

regularity – satisfied by any polynomial; not by sin/cos

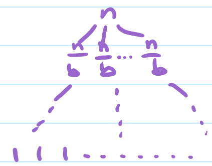
divide/combine at top dominates

Master Theorem

size of subproblems

Suppose $T(n) = \underbrace{a}_{\#\text{ of subproblems}} \cdot T\left(\frac{n}{b}\right) + \underbrace{f(n)}_{\text{divide+combine work}}$ $\quad \Theta(n^d)$

can be $\lfloor \frac{n}{b} \rfloor$, $\lceil \frac{n}{b} \rceil$, $\frac{n}{b} \pm c$



work

$f(n)$

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

$c \cdot n^{\log_b a}$

Then

$d < \log_b a$

if $f(n) \in O(n^{\log_b a - \varepsilon})$ for some $\varepsilon > 0$ then $T(n) \in \Theta(n^{\log_b a})$

base cases dominate

$d = \log_b a$

if $f(n) \in \Theta(n^{\log_b a})$ then $T(n) \in \Theta(n^{\log_b a} \cdot \log n)$

balanced

$d > \log_b a$

if $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ for some $\varepsilon > 0$ $\quad T(n) \in \Theta(f(n))$

and if $a \cdot f(\frac{n}{b}) \leq c \cdot f(n)$ for some $c > 1$ and all large n

regularity – satisfied by any polynomial; not by sin/cos

divide/combine at top dominates

$T(n) = 3 \cdot T(\lfloor \frac{n}{2} \rfloor) + n^1$ $\quad a=3,\ b=2,\ d=1$

$\log_b a = \log_2 3 \approx 1.585$ $\quad T(n)$ is $\Theta(n^{\log_2 3})$

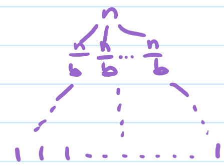
$1 < 1.585$

Master Theorem

size of subproblems

Suppose $T(n) = \underbrace{a}_{\text{\# of subproblems}} \cdot T\left(\frac{n}{b}\right) + \underbrace{f(n)}_{\text{divide+combine work}}$ $\quad \Theta(n^d)$

can be $\lfloor \frac{n}{b} \rfloor, \lceil \frac{n}{b} \rceil, \frac{n}{b} \pm c$



Then

$d < \log_b a$

if $f(n) \in O(n^{\log_b a - \varepsilon})$ for some $\varepsilon > 0$ then $T(n) \in \Theta(n^{\log_b a})$

base cases dominate

$d = \log_b a$

if $f(n) \in \Theta(n^{\log_b a})$ then $T(n) \in \Theta(n^{\log_b a} \cdot \log n)$

balanced

$d > \log_b a$

if $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ for some $\varepsilon > 0$ $T(n) \in \Theta(f(n))$

and if $a \cdot f(\frac{n}{b}) \leq c \cdot f(n)$ for some $c > 1$ and all large n

regularity – satisfied by any polynomial; not by sin/cos

divide/combine at top dominates

$T(n) = 1 \cdot T(\lfloor \frac{2}{3} n \rfloor) + 1^{=n^0}$ $\quad \log_b a = \log_{\frac{3}{2}} 1 = 0$ $\quad T(n) \in \Theta(n^0 \log n) = \Theta(\log n)$

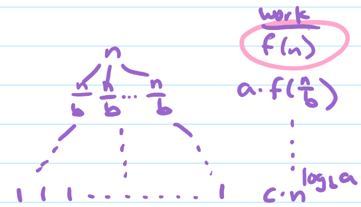
$a = 1 \quad b = \frac{3}{2} \quad d = 0$

$d = \log_b a$

Master Theorem

Suppose $T(n) = \underbrace{a}_{\text{\# of subproblems}} \cdot T(\overbrace{\tfrac{n}{b}}^{\text{size of subproblems}}) + \underbrace{f(n)}_{\text{divide+combine work}}$ $\Theta(n^d)$

can be $\lfloor \frac{n}{b} \rfloor, \lceil \frac{n}{b} \rceil, \frac{n}{b} \pm c$



Then

$d < \log_b a$

if $f(n) \in O(n^{\log_b a - \varepsilon})$ for some $\varepsilon > 0$ then $T(n) \in \Theta(n^{\log_b a})$

base cases dominate

$d = \log_b a$

if $f(n) \in \Theta(n^{\log_b a})$ then $T(n) \in \Theta(n^{\log_b a} \cdot \log n)$

balanced

$d > \log_b a$

if $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ for some $\varepsilon > 0$ $T(n) \in \Theta(f(n))$

and if $a \cdot f(\frac{n}{b}) \leq c \cdot f(n)$ for some $c > 1$ and all large n

divide/combine at top dominates

regularity – satisfied by any polynomial; not by sin/cos

$T(n) = 2 \cdot T(\lceil \frac{n}{4} \rceil) + T(\lfloor \frac{n}{4} \rfloor) + n \log n$ $\quad n \log n \in \Omega(n^{0.792 + 0.2})$

$a = 3 \quad b = 4$

$\log_b a = \log_4 3 \approx 0.792 \quad 3 \cdot \frac{n}{4} \log \frac{n}{4} = \frac{3}{4} n \cdot (\log n - \log 4) \leq n \log n$ for $n \geq 1$

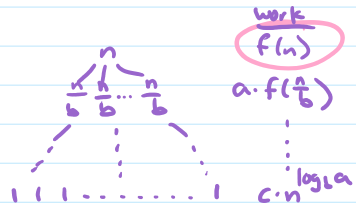
$T(n)$ is $\Theta(n \log n)$

Master Theorem

size of subproblems $\quad \Theta(n^d)$

Suppose $\quad T(n) = \underbrace{a}_{\#\text{ of subproblems}} \cdot T\left(\frac{n}{b}\right) + \underbrace{f(n)}_{\text{divide+combine work}}$

$\frac{n}{b}$ can be $\lfloor \frac{n}{b} \rfloor, \lceil \frac{n}{b} \rceil, \frac{n}{b} \pm c$



Then

$d < \log_b a$

if $f(n) \in O(n^{\log_b a - \varepsilon})$ for some $\varepsilon > 0$ then $T(n) \in \Theta(n^{\log_b a})$

base cases dominate

$d = \log_b a$

if $f(n) \in \Theta(n^{\log_b a})$ then $T(n) \in \Theta(n^{\log_b a} \cdot \log n)$

balanced

$d > \log_b a$

if $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ for some $\varepsilon > 0$ $\quad T(n) \in \Theta(f(n))$

and if $a \cdot f(\frac{n}{b}) \le c \cdot f(n)$ for some $c > 1$ and all large n

divide/combine at top dominates

regularity - satisfied by any polynomial; not by sin/cos

$T(n) = 2 \cdot T(\lceil \frac{n}{4} \rceil) + T(\lfloor \frac{n}{4} \rfloor) + n \log n \qquad n \log n \in \Omega(n^{0.792 + 0.2})$

$a = 3 \quad b = 4$

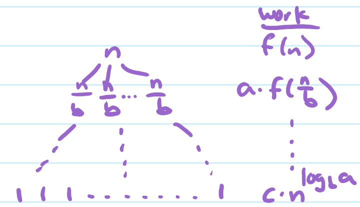
$\log_b a = \log_4 3 \approx 0.792 \qquad 3 \cdot \frac{n}{4} \log \frac{n}{4} = \frac{3}{4} n \cdot (\log n - \log 4) \le n \log n$ for $n \ge 1$

$T(n) \in \Theta(n \log n)$

Master Theorem

Suppose $T(n) = \underbrace{a}_{\text{\# of subproblems}} \cdot T\left(\overbrace{\tfrac{n}{b}}^{\text{size of subproblems}}\right) + \underbrace{f(n)}_{\text{divide+combine work}}$ $\quad \Theta(n^d)$

can be $\lfloor \frac{n}{b} \rfloor$, $\lceil \frac{n}{b} \rceil$, $\frac{n}{b} \pm c$



Then

$d < \log_b a$

if $f(n) \in O(n^{\log_b a - \varepsilon})$ for some $\varepsilon > 0$ then $T(n) \in \Theta(n^{\log_b a})$

base cases dominate

$d = \log_b a$

if $f(n) \in \Theta(n^{\log_b a})$ then $T(n) \in \Theta(n^{\log_b a} \cdot \log n)$

balanced

$d > \log_b a$

if $f(n) \in \Omega(n^{\log_b a + \varepsilon})$ for some $\varepsilon > 0$ $T(n) \in \Theta(f(n))$

and if $a \cdot f(\frac{n}{b}) \leq c \cdot f(n)$ for some $c < 1$ and all large n

divide/combine at top dominates

regularity – satisfied by any polynomial; not by sin/cos

$T(n) = T(\lceil \frac{n}{5} \rceil) + T(\lceil \frac{7}{10} n \rceil) + 101n$ Master Theorem does not apply (but $T(n)$ is $\Theta(n)$)