Analyzing Divide-and-Conquer Algs

Reminder

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

total time for divide and combine steps together

statements (time) when input is size n

of subproblems that we recursively solved

Size of each subproblem (really, $\lfloor n/b \rfloor$ or $\lceil \frac{n}{b} \rceil$)

Base: $T(c) = \Theta(1)$

problem size where you no longer recurse

split is $\Theta(1)$
merge is $\Theta(n)$

Merge sort

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

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

Divide-and-conquer closest pair

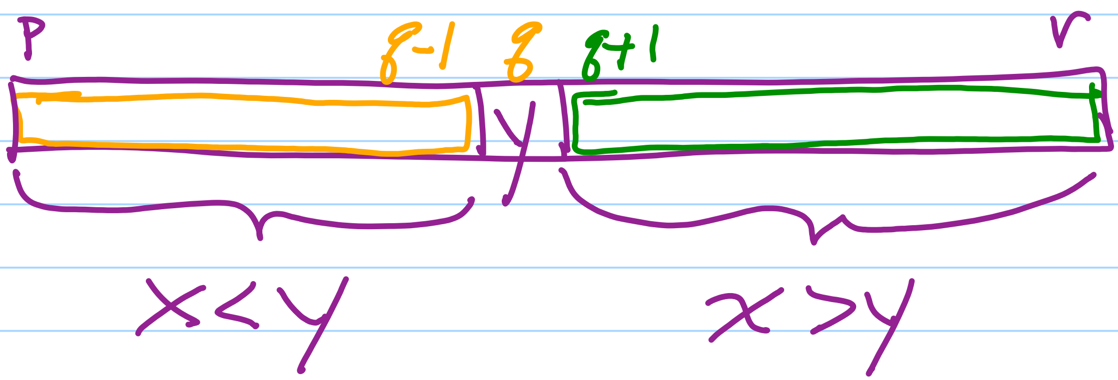
$a=2, \; b=2, \; f(n) = \Theta(n)$

both divide & combine
steps take linear time

Binary Search

Problem: Given a sorted array and a value x

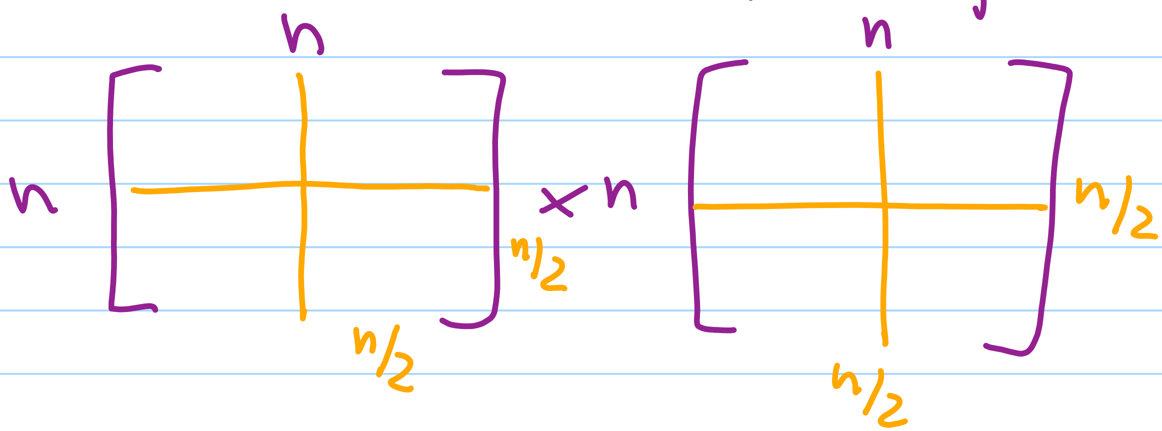
Asked if x is in the array



$$T(n) = T(n/2) + \Theta(1)$$

$a=1, \; b=2$

Strassen Matrix Multiplication



There are 7 multiplications of n/2 by n/2 matrices which can be combined in $\Theta(n^2)$ time to get the desired product

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

$a=7, \; b=2$

Master Method

Solve asymptotically any recurrence of form

$$T(n) = aT(n/b) + \Theta(n^{\ell} (\log n)^{k})$$

termination

$$T(c) = \Theta(1) \text{ for some constant } c$$

constants $a \geq 1$, $b > 1$, $\ell \geq 0$, $k \geq 0$

Look at $n^{\ell} (\log n)^{k}$

$f(n)$	ℓ	k
$\Theta(1)$	0	0
$\Theta(n)$	1	0
$\Theta(\sqrt{n})$	1/2	0
$\Theta(n \log n)$	1	1

Compare $\underbrace{l}_{n^l}$ and $\underbrace{\log_b a}_{n^{\log_b a}}$

$n^{\log_b a}$ is # of times termination condition is reached

Case 1 $\quad l < \log_b a$

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

Case 2 $\quad l = \log_b a$

$$T(n) = \Theta(f(n) \cdot \log n) = \Theta(n^l (\log n)^{k+1})$$
$$= \Theta(n^{\log_b a} (\log n)^{k+1})$$

$$T(n) = aT(n/b) + \Theta(n^{\ell}(\log n)^{k})$$

Case 3: $\ell > \log_b a$

$$T(n) = \Theta(f(n)) = \Theta(n^{\ell}(\log n)^{k})$$

Let's do some examples

$$T(n) = 2T(n/2) + \Theta(n) = \Theta(n \log n)$$

$a=2,\ b=2,\ \ell=1,\ k=0$

$\log_2 2 = 1 \qquad \log_b a = \ell$

$$T(n) = T(n/2) + \Theta(1) = \Theta(\log n)$$

$a=1$, $b=2$, $\ell=0$, $k=0$

$\log_b a = \log_2 1 = 0$ $\quad \ell = \log_b a$

$$T(n) = 7T(n/2) + \Theta(n^2) = \Theta(n^{\log_2 7})$$

$a=7$ $b=2$, $\ell=2$

$\log_b a = \log_2 7$ $\quad \log_b a > \ell$

Is $n^3 = \Theta(n^{3.000001})$? No!

$$\lim_{n \to \infty} \frac{n^{3.000001}}{n^3} = \infty$$

cost to sort to create ptsByYLeft + ptsByYRight

$$T(n) = 2T(n/2) + \Theta(\overline{n \log n})$$

$a=2, \ b=2 \quad \log_b a = 1$

$l=1 \ \ k=1 \qquad l = \log_b a$

$$\boxed{\Theta(n^l (\log n)^k)}$$

$$T(n) = \Theta(n(\log n)^2)$$

$n^{2.5}$

$$T(n) = 4T(n/2) + \underbrace{\Theta(n^2\sqrt{n})} = \Theta(n^{2.5})$$

$$= \Theta(n^2\sqrt{n})$$

$a=4, \ b=2 \ \ l=2.5, \ \ k=0$

$l > \log_b a$

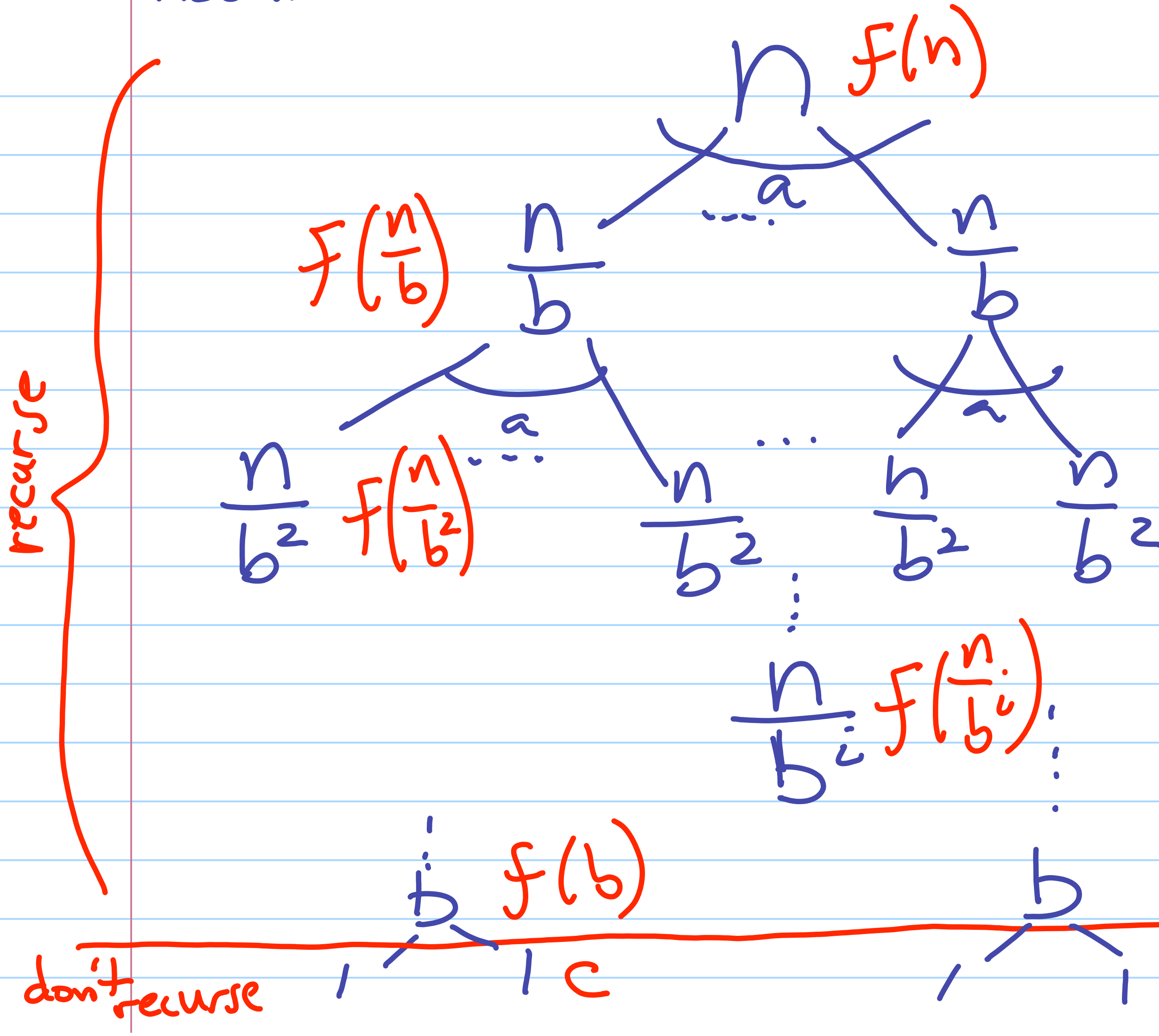
How can you exactly solve a recurrence of the form

$$T(1) = \underline{c}$$

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

where n is a power b

Recurrence Tree



level	#	time each
0	1 ×	$f(n)$
1	a ×	$f\left(\frac{n}{b}\right)$
2	a^2 ×	$f\left(\frac{n}{b^2}\right)$
i	a^i ×	$f\left(\frac{n}{b^i}\right)$
$(\log_b n)-1$		$f(b)$
$\log_b n$	$+\ a^{\log_b n}$ ×	c

level #	# nodes		time per node
0	$a^0 = 1$	$\times$	$f(n)$
1	$a^1 = a$	$\times$	$f(n/b)$
2	a^2	$\times$	$f(n/b^2)$
$\vdots$	$\vdots$	$\vdots$	
i	a^i	$\times$	$f(n/b^i)$
$\vdots$	$\vdots$		$\vdots$
$(\log_b n)-1$	$a^{(\log_b n)-1}$		
$\log_b n$	$a^{\log_b n}$	$\times$	c

recurse

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

verify by taking $\log_b$ of both sides

$$T(n) = \left[\sum_{i=0}^{(\log_b n)-1} a^i \times f(n/b^i) \right] + n^{\log_b a} \cdot c$$

Master Method $F(n) = \Theta(n^{\ell} (\log n)^{k})$

$$T(n) = \left[\sum_{i=0}^{(\log_b n)-1} a^i \, \Theta\left(\left(\frac{n}{b^i}\right)^{\ell} \cdot \left(\log \frac{n}{b^i}\right)^{k} \right) \right] + n^{\log_b a} \cdot c$$

$$\log \frac{n}{b^i} = \log n - \log b^i$$
$$= \log n - i \log b$$

$b=2$
$a=2$

$T(1)=1$

$$T(n)=2T(n/2)+\underbrace{cn}_{f(n)}$$

$$T(n)=\left[\sum_{i=0}^{(\log_b n)-1} a^i f(n/b^i)\right]+\overset{T(1)}{1}\cdot n^{\log_b a}$$

$$=\left[\sum_{i=0}^{(\log_2 n)-1} \cancel{2^i}\cdot c\cdot\frac{n}{\cancel{2^i}}\right]+n$$

$$=cn\cdot\log_2 n+n$$

exact for n a power of 2

$T(1)=1,\quad T(n)=4T(n/2)+cn$

$a=4,\ b=2$
$\log_b a=2$

$$T(n)=\left[\sum_{i=0}^{(\log_b n)-1} a^i f(n/b^i)\right]+1\cdot n^{\log_b a}$$

$$=\left[\sum_{i=0}^{(\log_2 n)-1} 4^i \frac{c\cdot n}{2^i}\right]+n^2$$

$$=\left(cn\sum_{i=0}^{(\log_2 n)-1} 2^i\right)+n^2 = cn\left(2^{\log_2 n}-1\right)+n^2$$

n

$$= cn(n-1)+n^2 = \boxed{(c+1)n^2-cn}$$

geometric sum $\displaystyle\sum_{i=0}^{m} x^i=\frac{x^{m+1}-1}{x-1}$ $\quad x\neq 1$

Does $T(n) = 2T(n/2) + \Theta(n/\log n)$ fit into master method?

No since we would need to set k to -1 & there's a restriction that $k \geq 0$.

$$T(n) = T(n-1) + 10, \quad \underline{T(1) = 1}$$

recurse

n	10	how many
$n-1$	10	
$n-2$	10	
$\vdots$		
2	10	

terminate

1	$+1$

$$T(n) = 10(n-2+1) + 1$$

$$= 10n - 10 + 1$$

$$\boxed{= 10n - 9}$$

Aside

\# of numbers $x, x+1, \ldots, y$

$$y - x + 1$$

$T(n) = T(n-1) + 10, \quad T(1) = 1$

Claim $T(n) = 10n - 9$

Sanity check

	$10n-9$
$T(1) = 1$	$10 - 9 = 1$ ✓
$T(2) = T(1) + 10 = 11$	$20 - 9 = 11$ ✓
$T(3) = T(2) + 10 = 21$	$30 - 9 = 21$ ✓

You can prove correctness using mathematical induction