CMSC351-2023-10-02

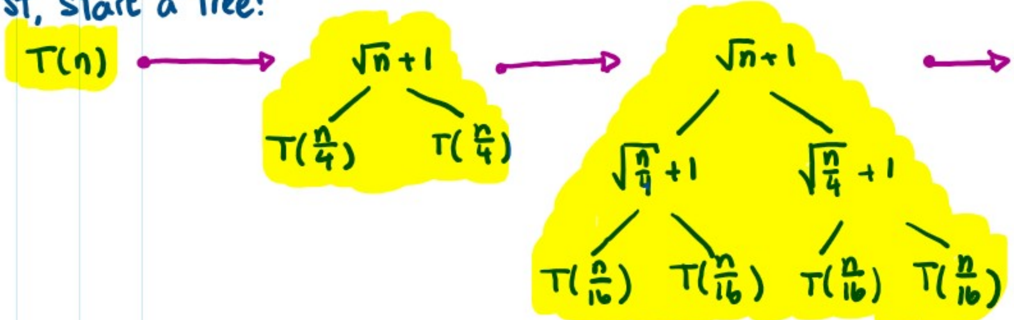
Monday, October 2, 2023 8:29 AM

ONE MORE RECURRENCE TREE EXAMPLE!

$T(☺) = 2T(\frac{☺}{4}) + \sqrt{☺} + 1$ for any ☺ except ☺ = 1

Consider $T(n) = 2T(\frac{n}{4}) + \sqrt{n} + 1$ w/ $T(1) = 3$

First, start a tree!



⋮ etc

Looking at the tree leaves, they are, progressively, $T(n/4^i)$ where i = level (root is $i=0$).

Ends when $n/4^i = 1$ b/c $T(1)=3$. Thats $n = 4^i$ or $i = \log_4 n$ ← Leaf level!

Now build a table:

Level	# per level	time/node	level time
0	1	$\sqrt{n}+1$	$1(\sqrt{n}+1) = \sqrt{n}+1$
1	2	$\sqrt{\frac{n}{4}}+1$	$2(\sqrt{\frac{n}{4}}+1) = \sqrt{n}+2$
2	4	$\sqrt{\frac{n}{16}}+1$	$4(\sqrt{\frac{n}{16}}+1) = \sqrt{n}+4$
3	8	$\sqrt{\frac{n}{64}}+1$	$8(\sqrt{\frac{n}{64}}+1) = \sqrt{n}+8$
⋮	⋮	⋮	
$\log_4 n - 1$	$2^{\log_4 n - 1}$		$\sqrt{n} + 2^{\log_4 n - 1}$
$\log_4 n$	$2^{\log_4 n}$	3	$3 \cdot 2^{\log_4 n}$

see pattern!

b/c leaves are $T(1)=3$

leaves special!

Before we sum, observe, by change of base:

$$\log_4 n = \frac{\lg n}{\lg 4} = \frac{\lg n}{2} = \frac{1}{2}\lg n = \lg \sqrt{n}$$

so $2^{\log_4 n} = 2^{\lg \sqrt{n}} = \sqrt{n}$

and $2^{\log_4 n - 1} = 2^{-1} 2^{\log_4 n} = \frac{1}{2}\sqrt{n}$

So now total time (sum of final column) is:

$$T(n) = \underbrace{3\sqrt{n}}_{\text{Leaves}} + \underbrace{\sum_{i=0}^{\log_4 n - 1} (\sqrt{n} + 2^i)}_{\text{non-leaves}} = 3\sqrt{n} + \underbrace{\sum_{i=0}^{\log_4 n - 1} \sqrt{n}}_{\text{Ez!}} + \underbrace{\sum_{i=0}^{\log_4 n - 1} 2^i}_{\text{geometric}}$$

Leaves   non-leaves   Ez!   geometric

THE MASTER THEOREM!

① **Premise:** Most recurrence relns we've done have this general form:

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

Q: Could we perhaps find a general rule for this?

A: YES! :)

② **The Master Theorem:** Sps we have $T(n) = aT(\frac{n}{b}) + f(n)$ with $a, b \in \mathbb{Z}^+$ w/ $b > 1$.

note $\lg^k n = (\lg n)^k$

Then we have 3(+1) possible cases:

(i) if $f(n) = O(n^c)$ and $\log_b a > c$ then $T(n) = \Theta(n^{\log_b a})$.

(ii) if $f(n) = \Theta(n^c)$ and $\log_b a = c$ then $T(n) = \Theta(n^{\log_b a} \lg n)$

(ii)(f) if $f(n) = \Theta(n^c \lg^k n)$ and $\log_b a = c$ then $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$

(iii) if $f(n) = \Omega(n^c)$ and $\log_b a < c$ then $T(n) = \Theta(f(n))$

note Case (iii) also requires a regularity condition on f. This condition will be satisfied for all f(n) we encounter so don't worry about it!

Comment: to be systematic, a good approach is to step through the Θ, O, Ω nature of f(n) and see, in turn, if the cases apply. Easiest (as we'll see) to do cases (ii), (ii)(f) first.

ex $T(n) = 4T(n/2) + n^2 + \lg n$

First observe $a=4, b=2$. $\log_2 4 = 2$. And $f(n) = n^2 + \lg n$

Now then:

- $f(n) = \Theta(n^2)$ so $c = 2$

  Observe: $\log_2 4 = 2 = c$ so (ii) applies and $\underbrace{T(n) = \Theta(n^{\log_2 4} \lg n) = \Theta(n^2 \lg n)}_{\text{DONE!}}$

ex $T(n) = 3T(\frac{n}{4}) + n\lg n + 1$

First observe $a=3, b=4$, $\log_4 3 \approx 0.*$. And $f(n) = n\lg n + 1$

Now then: $n^1 \lg n$

- $f(n) = \Theta(n \lg n)$ so $c = 1$

Now then: $\curvearrowleft n^1 \lg n$

- $f(n) = \Theta(n \lg n)$ so $c = 1$

  observe: $\log_4 3 \approx 0.* \neq 1 = c$ so not case (ii)

- $f(n) = O(n^2)$ so $c = 2$

  observe: $\log_4 3 \approx 0.* \not> 2 = c$ so not case (i)

- $f(n) = \Omega(n)$ so $c = 1$

  observe: $\log_4 3 \approx 0.* < 1 = c$ so case (iii) applies

  and so $T(n) = \Theta(f(n)) = \Theta(n \lg n + 1) = \Theta(n \lg n)$

DONE!