COMP 482 / ELEC 420

Recurrences

Math Background: Review & Beyond

1. Asymptotic concepts & useful math
2. Recurrences
3. Probabilistic analysis

Your To-Do List

- Read [CLRS] 4.

- Assignment 2.

Obtaining Recurrences

Introductory multiplication examples:

$T(n) = k$ $n=1$
$T(n) = 4T(n/2) + kn$ $n>1$

$T'(n) = k'$ $n=1$
$T'(n) = 3T'(n/2) + k'n$ $n>1$

Obtained from straightforward reading of algorithms.

Key observation: Most deterministic algorithms lead to such recurrences.

1. Determine appropriate metric for the size "n".
   - Usually obvious, but will see exceptions.
2. Examine algorithm's recursion/iteration & how problem sizes change.
   - Usually straightforward.

Solving Recurrences

T(n) = k	n=1
T(n) = 4T(n/2) + kn	n>1

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

T'(n) = k'	n=1
T'(n) = 3T'(n/2) + k'n	n>1

$$T'(n) = \Theta\left(n^{\log_2 3}\right)$$

Real goal: Find closed form of defined functions.

<u>How?</u>

In general, hard. Solutions not always known.

Will discuss techniques in a few minutes…

First look at a couple side issues.

Recurrences

$$T(n) = k \qquad n=1$$
$$T(n) = 4T(n/2) + kn \qquad n>1$$

$$T'(n) = k' \qquad n=1$$
$$T'(n) = 3T'(n/2) + k'n \qquad n>1$$

? What if n is odd? ?

Next iteration, n is not integral. Nonsense.

$$T(n) = k \qquad n=1$$
$$T(n) = 3T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + kn \qquad n>1$$

Above more accurate.

The difference rarely matters, so usually ignore this detail.

Recurrences

$$\begin{array}{ll} T(n) = k & n=1 \\ T(n) = 4T(n/2) + kn & n>1 \end{array}$$

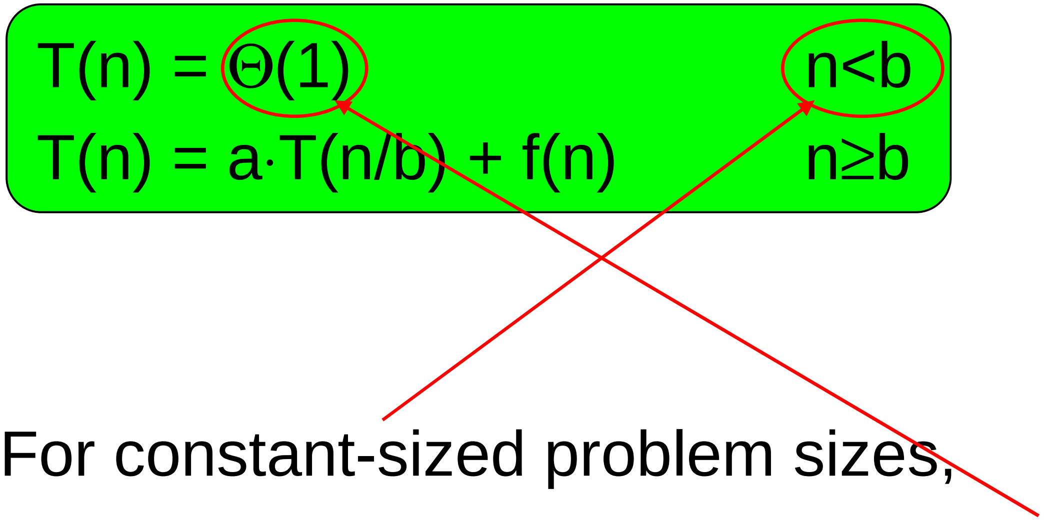
$$\begin{array}{ll} T'(n) = k' & n=1 \\ T'(n) = 3T'(n/2) + k'n & n>1 \end{array}$$

Both are examples of the most common form.

Divide-and-conquer, with each step dividing into “a” equal-size subproblems.

$$\begin{array}{ll} T(n) = O(g(n)) & n<b \\ T(n) = a \cdot T(n/b) + f(n) & n \geq b \end{array}$$

Recurrences



$$\begin{array}{ll} T(n) = \Theta(1) & n<b \\ T(n) = a \cdot T(n/b) + f(n) & n \geq b \end{array}$$

For constant-sized problem sizes,
can bound algorithm by some constant value.

This constant value is irrelevant for asymptote.
Thus, often skip writing the base case equation.

Techniques for Solving Recurrences

We'll use four techniques:

- Substitution
- Recursion Tree

- Master Method – for divide & conquer

- Characteristic Equation – for linear

Techniques: Substitution

Guess a solution & check it.

More detail:

1. Guess the form of the solution, using unknown constants.
2. Use induction to find the constants & verify the solution.

Completely dependent on making reasonable guesses.

Substitution Example 1

$$T(n) = \begin{cases} 1 & n=1 \\ 4T(n/2) + n & n>1 \end{cases}$$

Simplified version of previous example.

Guess: $T(n) = O(n^3)$.

More specifically:

$T(n) \leq cn^3$, for all large enough n.

Substitution Example 1

$$\begin{aligned} &T(n) = 1 && n=1 \\ &T(n) = 4T(n/2) + n && n>1 \end{aligned}$$

Guess:

$T(n) \le cn^3$ for $\forall n>n_0$

Prove by strong induction on n.

? Which means what exactly? ?

Assume: $T(k) \le ck^3$ for $\forall k>n_0$, for $\forall k<n$.

Show: $T(n) \le cn^3$ for $\forall n>n_0$.

Substitution Example 1

$$T(n) = 1 \qquad n=1$$

$$T(n) = 4T(n/2) + n \qquad n>1$$

Guess:

$T(n) \le cn^3$ for $\forall n>n_0$

Assume $T(k) \le ck^3$ for $\forall k>n_0$, for $\forall k<n$. Show $T(n) \le cn^3$ for $\forall n>n_0$.

Base case, $n=n_0+1$:

Awkward. Fortunately, $n_0=0$ works in these examples.

Substitution Example 1

$T(n) = 1$	$n=1$
$T(n) = 4T(n/2) + n$	$n>1$

Guess:
$T(n) \le cn^3$

Assume $T(k) \le ck^3$, for $\forall k<n$. Show $T(n) \le cn^3$.

Base case, n=1:

$T(n) = 1$	Definition.
$1 \le c$	Choose large enough c for conclusion.

Substitution Example 1

$$T(n) = \begin{cases} 1 & n=1 \\ 4T(n/2) + n & n>1 \end{cases}$$

Guess:
$T(n) \le cn^3$

Assume $T(k) \le ck^3$, for $\forall k<n$. Show $T(n) \le cn^3$.

Inductive case, $n>1$:

$T(n) = 4T(n/2) + n$	Definition.
$\le 4c\cdot(n/2)^3 + n$	Induction.
$= c/2 \cdot n^3 + n$	Algebra.

While this is $O(n^3)$, we're not done.

Need to show $c/2 \cdot n^3 + n \le c\cdot n^3$.

Fortunately, the constant factor is shrinking, not growing.

Substitution Example 1

$$T(n) = 1 \qquad n=1$$
$$T(n) = 4T(n/2) + n \qquad n>1$$

Guess:
$T(n) \leq cn^3$

Assume $T(k) \leq ck^3$, for $\forall k<n$. Show $T(n) \leq cn^3$.

Inductive case, $n>1$:

$T(n) \leq c/2 \cdot n^3 + n$	From before.
$= cn^3 - (c/2 \cdot n^3 - n)$	Algebra.
$\leq cn^3$	Since $n>0$, if $c \geq 2$.

Substitution Example 1

$T(n) = 1$ $n=1$

$T(n) = 4T(n/2) + n$ $n>1$

Proved:

$T(n) \leq 2n^3$ for $\forall n>0$

Thus, $T(n) = O(n^3)$.

Substitution Example 2

$$T(n) = 1 \qquad n=1$$

$$T(n) = 4T(n/2) + n \qquad n>1$$

Guess: $T(n) = O(n^2)$.

Same recurrence, but now try tighter bound.

More specifically:

$T(n) \leq cn^2$ for $\forall n>n_0$.

Substitution Example 2

$$T(n) = \begin{cases} 1 & n=1 \\ 4T(n/2) + n & n>1 \end{cases}$$

Guess:

$T(n) \le cn^2$ for $\forall n>n_0$

Follow same steps, and we get...

Assume $T(k) \le ck^2$, for $\forall k<n$. Show $T(n) \le cn^2$.

$$\begin{aligned} T(n) &= 4T(n/2) + n \\ &\le 4c\cdot(n/2)^2 + n \\ &\boxed{= cn^2 + n} \end{aligned}$$

Not $\le cn^2$!

Problem is that the constant isn't shrinking.

Substitution Example 2

$$T(n) = \begin{cases} 1 & n=1 \\ 4T(n/2) + n & n>1 \end{cases}$$

Solution: Use a tighter guess & inductive hypothesis.

Subtract a lower-order term – a common technique.

Guess:
$T(n) \leq cn^2 - dn$ for $\forall n>0$

Assume $T(k) \leq ck^2 - dk$, for $\forall k<n$. Show $T(n) \leq cn^2 - dn$.

Substitution Example 2

$T(n) = 1$	$n=1$
$T(n) = 4T(n/2) + n$	$n>1$

Guess:
$T(n) \le cn^2 - dn$

Assume $T(k) \le ck^2 - dn$, for $\forall k<n$. Show $T(n) \le cn^2 - dn$.

Base case, n=1:

$T(n) = 1$	Definition.
$1 \le c-d$	Choosing c, d appropriately.

Substitution Example 2

$T(n) = 1$	$n=1$
$T(n) = 4T(n/2) + n$	$n>1$

Guess:
$T(n) \leq cn^2 - dn$

Assume $T(k) \leq ck^2 - dn$, for $\forall k<n$. Show $T(n) \leq cn^2 - dn$.

Inductive case, n>1:

$T(n) = 4T(n/2) + n$	Definition.
$\leq 4(c(n/2)^2 - d(n/2)) + n$	Induction.
$= cn^2 - 2dn + n$	Algebra.
$= cn^2 - dn - (dn - n)$	Algebra.
$\leq cn^2 - dn$	Choosing $d \geq 1$.

Substitution Example 2

$$T(n) = 1 \qquad n=1$$

$$T(n) = 4T(n/2) + n \qquad n>1$$

Proved:

$T(n) \leq 2n^2 - 1n$ for $\forall n>0$

Thus, $T(n) = O(n^2)$.

Substitution Summary

Ability to guess effectively comes with experience.

Examples used O(), guessing T() ≤ …

Can use Ω(), guessing T() ≥ …

Techniques: Recursion Tree

1. Unroll the recurrence to obtain a summation.
2. Solve or estimate summation.
3. Use solution as a guess in substitution.

Math can be tricky.

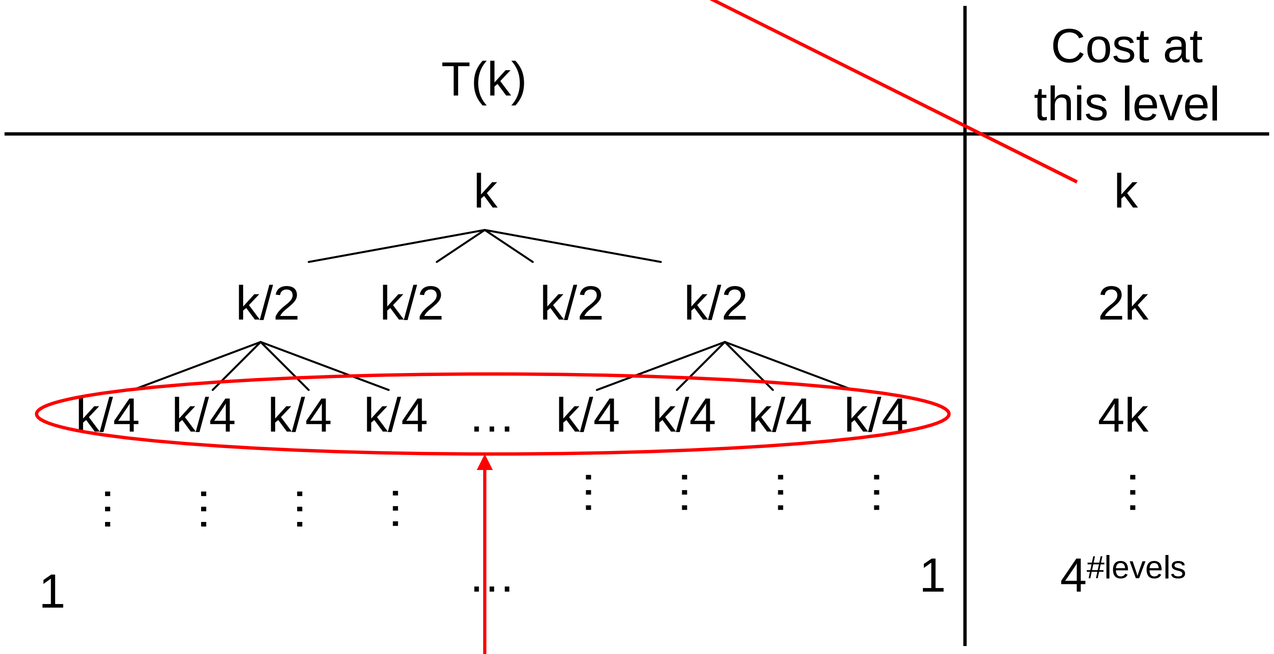
Recursion Tree Example 1

$$T(n) = 1 \qquad n=1$$

$$T(n) = 4T(n/2) + n \qquad n>1$$

? How many levels? ?

$\log_2 k$



T(k)	Cost at this level
k	k
k/2 k/2 k/2 k/2	2k
k/4 k/4 k/4 k/4 … k/4 k/4 k/4 k/4	4k
⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮ ⋮	⋮
1 … 1	$4^{\#levels}$

Now, turn picture into a summation…

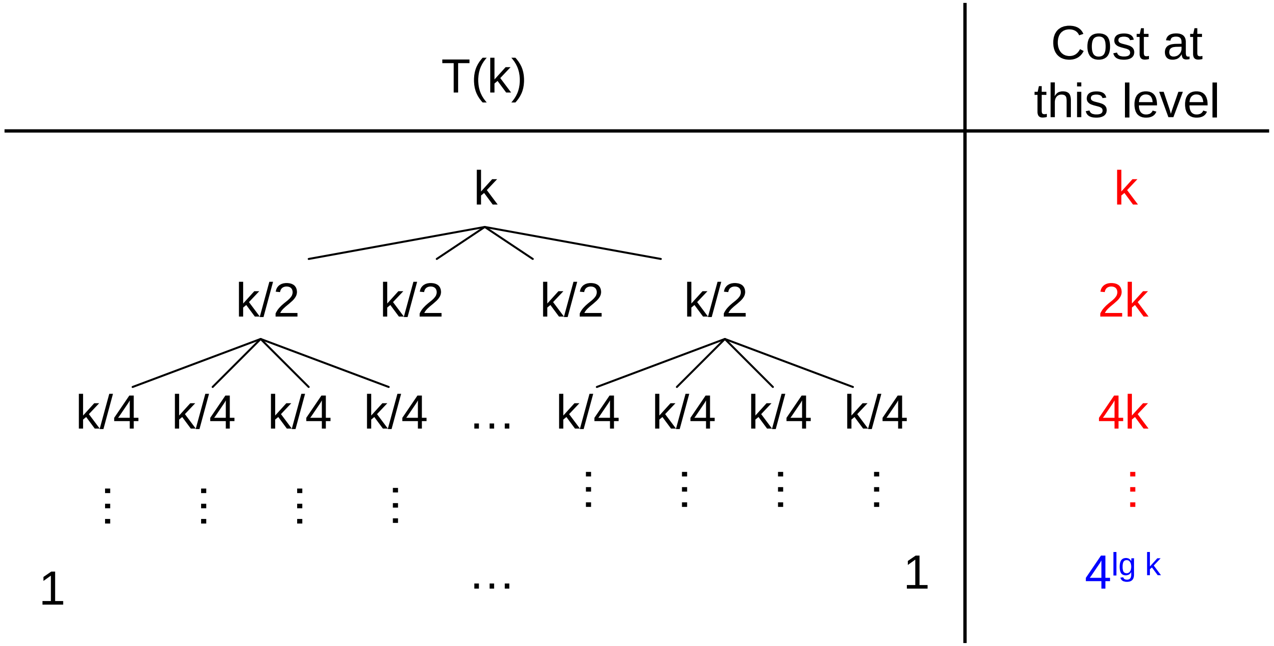
In this example, all terms on a level are the same.

Common, but not always true.

Recursion Tree Example 1

$T(n) = 1 \qquad n=1$

$T(n) = 4T(n/2) + n \qquad n>1$



$$T(k) = k + 2k + 4k + \ldots + 2^{\lg k - 1}k + 4^{\lg k}$$

$$= k(1 + 2 + 4 + \ldots + 2^{\lg k - 1}) + k^{\lg 4}$$

$$T(k) = k\left(\sum_{i=0}^{\lg k-1} 2^i\right) + k^2$$

$$= k\left(\frac{2^{\lg k-1}}{2-1}\right) + k^2$$

$$= k\left(\frac{2^{\lg k}}{2}\right) + k^2$$

$$= k\left(\frac{k^{\lg 2}}{2}\right) + k^2$$

$$= k\left(\frac{k}{2}\right) + k^2$$

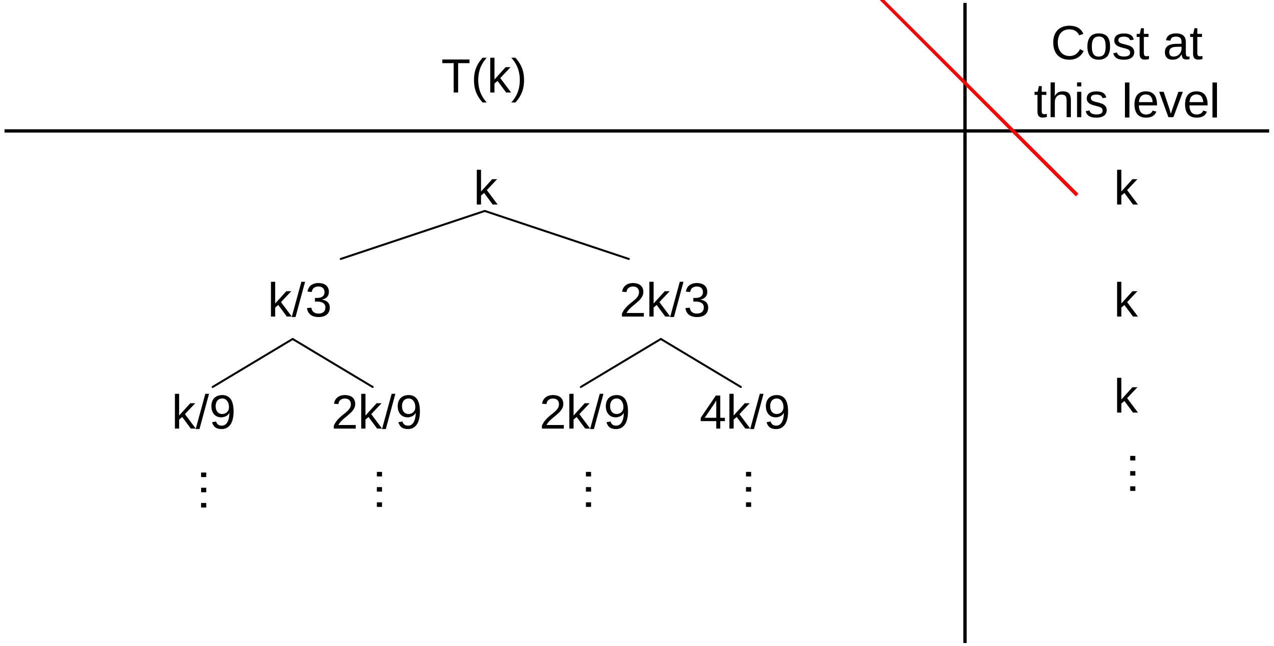
$$= \Theta(k^2)$$

Recursion Tree Example 2

$$\begin{array}{ll} T(n) = 1 & n=1 \\ T(n) = T(n/3) + T(2n/3) + n & n>1 \end{array}$$

? How many levels? ?

$\log_{3/2} k$



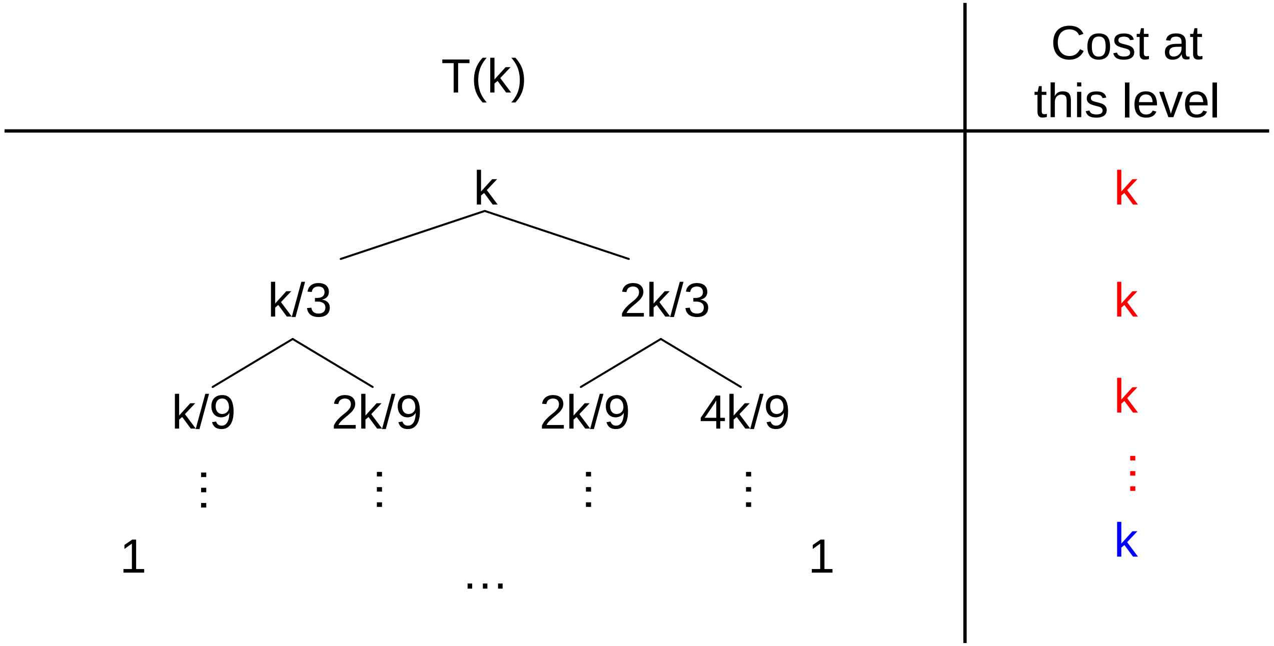
But, not all branches have same depth!

Makes cost near the leaves hard to calculate.

Estimate!

Recursion Tree Example 2

$$T(n) = 1 \qquad n=1$$
$$T(n) = T(n/3) + T(2n/3) + n \qquad n>1$$



#levels = $\log_{3/2} k$

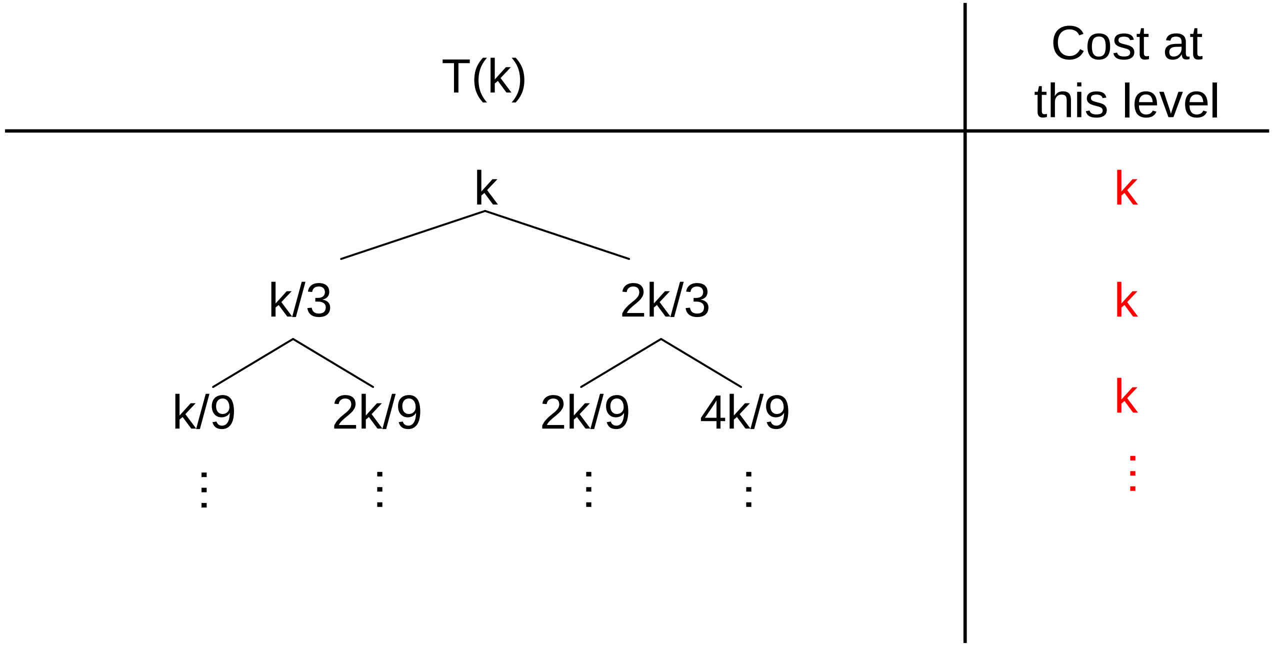
<u>Overestimate.</u>

Consider all branches to be of max depth.

$$T(k) \le k \cdot (\log_{3/2} k - 1) + k$$
$$T(k) = O(k \log k)$$

Recursion Tree Example 2

$$T(n) = \begin{cases} 1 & n=1 \\ T(n/3) + T(2n/3) + n & n>1 \end{cases}$$



T(k)

Cost at this level

k

k/3 2k/3

k/9 2k/9 2k/9 4k/9

⋮ ⋮ ⋮ ⋮

k

k

k

⋮

<u>Underestimate.</u>

Count the complete levels, & ignore the rest.

$T(k) \geq k \cdot (\log_3 k - 1)$

$T(k) = \Omega(k \log k)$

#levels = $\log_3 k$

Thus, $T(n) = \Theta(n \log n)$

Recursion Tree Example 3

$$T(n) = 1 \qquad n=1$$
$$T(n) = T(n-1) + n \qquad n>1$$

$$\begin{aligned} T(k) &= k + T(k-1) \\ &= k + k-1 + T(k-2) \\ &= k + k-1 + k-2 + T(k-3) \\ &= \ldots \\ &= k + k-1 + k-2 + \ldots + 2 + 1 \\ &= \sum_{i=1}^{k} i = \frac{k(k+1)}{2} = O\left(k^2\right) \end{aligned}$$

Linear recurrences with only one recurrence are relatively easy.

Techniques: Master Method

Cookbook solution for <u>some</u> recurrences of the form

$$T(n) = a \cdot T(n/b) + f(n)$$

where

$a \geq 1$, $b > 1$, $f(n)$ asymptotically positive

First describe its cases, then outline proof.

Master Method Case 1

$$T(n) = a \cdot T(n/b) + f(n)$$

$$f(n) = O(n^{\log_b a - \varepsilon}) \text{ for some } \varepsilon>0 \;\rightarrow\; T(n) = \Theta(n^{\log_b a})$$

$T(n) = 7T(n/2) + cn^2$ $\quad$ $a=7,\ b=2$

E.g., Strassen matrix multiplication.

$cn^2 \overset{?}{=} O(n^{\log_b a - \varepsilon}) = O(n^{\log_2 7 - \varepsilon}) \approx O(n^{2.8 - \varepsilon})$

Yes, for any $\varepsilon \leq 0.8$.

$T(n) = \Theta(n^{\lg 7})$

Master Method Case 2

$$T(n) = a \cdot T(n/b) + f(n)$$

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

$T(n) = 2T(n/2) + cn$ $a=2, b=2$

E.g., mergesort.

$cn =^? \Theta(n^{\log_b a}) = \Theta(n^{\log_2 2}) = \Theta(n)$

Yes.

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

Master Method Case 3

$$T(n) = a \cdot T(n/b) + f(n)$$

$f(n) = \Omega(n^{\log_b a + \varepsilon})$ for some $\varepsilon>0$ and

$a \cdot f(n/b) \leq c \cdot f(n)$ for some $c<1$ and all large enough n

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

I.e., is the
constant factor
shrinking?

$T(n) = 4 \cdot T(n/2) + n^3$ $\qquad$ $a=4,\ b=2$

$n^3 =^? \Omega(n^{\log_b a + \varepsilon}) = \Omega(n^{\log_2 4 + \varepsilon}) = \Omega(n^{2 + \varepsilon})$

Yes, for any $\varepsilon \leq 1$.

$4(n/2)^3 = \frac{1}{2} \cdot n^3 \leq^? cn^3$

Yes, for any $c \geq \frac{1}{2}$. $\qquad$ $T(n) = \Theta(n^3)$

Master Method Case 4

$T(n) = a \cdot T(n/b) + f(n)$

None of previous apply. Master method doesn't help.

$T(n) = 4T(n/2) + n^2/\lg n$ a=4, b=2

Case 1?

$n^2/\lg n \overset{?}{=} O(n^{\log_b a - \varepsilon}) = O(n^{\log_2 4 - \varepsilon}) = O(n^{2 - \varepsilon}) = O(n^2/n^{\varepsilon})$

No, since lg n is asymptotically less than n^{ε}.

Thus, $n^2/\lg n$ is asymptotically greater than n^2/n^{ε}.

Master Method Case 4

$$T(n) = a \cdot T(n/b) + f(n)$$

None of previous apply. Master method doesn't help.

$T(n) = 4T(n/2) + n^2/\lg n$ a=4, b=2

Case 2?

$n^2/\lg n \overset{?}{=} \Theta(n^{\log_b a}) = \Theta(n^{\log_2 4}) = \Theta(n^2)$

No.

Master Method Case 4

$T(n) = a \cdot T(n/b) + f(n)$

None of previous apply. Master method doesn't help.

$T(n) = 4T(n/2) + n^2/\lg n$ a=4, b=2

Case 3?

$n^2/\lg n \overset{?}{=} \Omega(n^{\log_b a + \varepsilon}) = \Omega(n^{\log_2 4 + \varepsilon}) = \Omega(n^{2 + \varepsilon})$

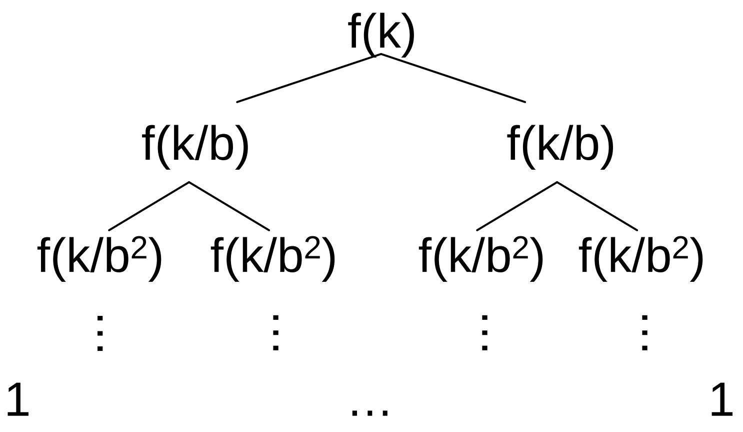
No, since $1/\lg n$ is asymptotically less than n^ε.

Master Method Proof Outline

$T(n) = a \cdot T(n/b) + f(n)$

How many levels?

$\log_b k$



T(k)	Cost at this level
$f(k)$	$f(k)$
$f(k/b)$ $f(k/b)$	$a \cdot f(k/b)$
$f(k/b^2)$ $f(k/b^2)$ $f(k/b^2)$ $f(k/b^2)$	$a^2 \cdot f(k/b^2)$
⋮ ⋮ ⋮ ⋮	⋮
1 … 1	$a^{\#levels}$

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

Cases correspond to determining which term dominates & how to compute sum.

Techniques: Characteristic Equation

Applies to linear recurrences:

- Homogenous:

  $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k}$

- Nonhomogenous:

  $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k} + F(n)$

Homogenous – With Example

$a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k}$

$a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$

$a_0 = 2$
$a_1 = 5$
$a_2 = 15$

Guess $a_n = r^n$.
(Not necessarily true, but leads to a sol'n.)
Plug in.

$r^n = c_1r^{n-1} + c_2r^{n-2} + \ldots + c_kr^{n-k}$

$r^n = 6r^{n-1} - 11r^{n-2} + 6r^{n-3}$

Divide by r^{n-k}.
Move terms.
"Characteristic Equation"

$r^k - c_1r^{k-1} - c_2r^{k-2} - \ldots - c_k = 0$

$r^3 - 6r^2 + 11r - 6 = 0$

Homogenous – With Example

$r^k - c_1r^{k-1} - c_2r^{k-2} - \ldots - c_k = 0$

$r^3 - 6r^2 + 11r - 6 = 0$

$= (r-1)(r-2)(r-3)$

Find roots.
(We'll assume distinct.)

$r_1, r_2, \ldots, r_k$

$r_1=1, r_2=2, r_3=3$

Plug into
this eq'n:

$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \ldots + \alpha_k r_k^n$

$a_n = \alpha_1 + \alpha_2 2^n + \alpha_3 3^n$

Use initial
conditions
to solve.

$a_0 = 2 = \alpha_1+\alpha_2+\alpha_3$

$a_1 = 5 = \alpha_1+\alpha_2 2+\alpha_3 3$

$a_2 = 15 = \alpha_1+\alpha_2 4+\alpha_3 9$

$\alpha_1, \alpha_2, \ldots, \alpha_k$

$\alpha_1=1, \alpha_2=-1, \alpha_3=2$

$a_n = 1 - 2^n + 2 \cdot 3^n$

Let’s Verify This Works

Assume $r_1, r_2, \ldots, r_k$ are roots of

$r^k - c_1r^{k-1} - c_2r^{k-2} - \ldots - c_k = 0$.

$a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k}$

$\leftrightarrow$

$a_n = \alpha_1r_1^n + \alpha_2r_2^n + \ldots + \alpha_kr_k^n$

General proof omitted.

$a_0 = 2, a_1 = 5, a_2 = 15$

$\forall n \geq 3$,

$a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$

$\leftrightarrow$

$a_n = 1 - 2^n + 2 \cdot 3^n$

Strong induction on n.

Let’s Verify This Works

$a_0 = 2, a_1 = 5, a_2 = 15$

$\forall n \geq 3,$

$a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$

$\leftrightarrow$

$a_n = 1 - 2^n + 2 \cdot 3^n$

Base, n=3:

$a_3 = 6a_2 - 11a_1 + 6a_0 = 90-55+12 = \mathbf{47}$

$a_3 = 1-2^3+2 \cdot 3^3 = 1-8+54 = \mathbf{47}$

Inductive: Assume $\forall k \leq n$

$a_k = 6a_{k-1} - 11a_{k-2} + 6a_{k-3}$

$a_k = 1 – 2^k + 2 \cdot 3^k$

Show

$6a_n-11a_{n-1}+6a_{n-2} = \mathbf{1-2^{n+1}+2 \cdot 3^{n+1}}$

$6a_n-11a_{n-1}+6a_{n-2} =$

$6(1 - 2^n + 2 \cdot 3^n)-11(1 - 2^{n-1} + 2 \cdot 3^{n-1})+6(1 - 2^{n-2} + 2 \cdot 3^{n-2}) =$

$(6-11+6) – (6 \cdot 2^n-11 \cdot 2^{n-1}+6 \cdot 2^{n-2}) + (12 \cdot 3^n-22 \cdot 3^{n-1}+12 \cdot 3^{n-2}) =$

$1 – (6 \cdot 2^n-8 \cdot 2^{n-1}) + (12 \cdot 3^n-18 \cdot 3^{n-1}) =$

$1 - (2 \cdot 2^n) + (6 \cdot 3^n) = \mathbf{1-2^{n+1}+2 \cdot 3^{n+1}}$

Nonhomogenous – With Example

$a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k} + F(n)$

$a_n = 3a_{n-1} + 2n$ $\quad a_0 = 1/3$

Solve as before, ignoring F(n).

$r^n = c_1r^{n-1} + c_2r^{n-2} + \ldots + c_kr^{n-k}$

$r^n = 3r^{n-1}$

$r^k - c_1r^{k-1} - c_2r^{k-2} - \ldots - c_k = 0$

$r - 3 = 0$

$r_1, r_2, \ldots, r_k$

$r_1=3$

$a_n = \alpha_1r_1^{\,n} + \alpha_2r_2^{\,n} + \ldots + \alpha_kr_k^{\,n}$

$a_n = \alpha_1 3^n$

This is only a partial solution.

Nonhomogenous – With Example

$a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k} + F(n)$

$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \ldots + \alpha_k r_k^n$

$a_n = 3a_{n-1} + 2n$ $a_0 = 1/3$

$a_n = \alpha_1 3^n$

Now look at F(n)

F(n)

F(n) = 2n

Guess a solution of similar form, & solve.

a_n = ...something like F(n)…

$a_n = cn + d$

$cn + d = 3(c(n-1)+d) + 2n$

$(2+2c)n + (2d-3c) = 0$

$c = -1, d = -3/2$

$a_n = -n - 3/2$

This is another partial solution.

Nonhomogenous – With Example

$a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k} + F(n)$

$a_n = \alpha_1r_1^n + \alpha_2r_2^n + \ldots + \alpha_kr_k^n$

$a_n = \ldots$

$a_n = 3a_{n-1} + 2n$ $\qquad a_0 = 1/3$

$a_n = \alpha_13^n$

$a_n = -n - 3/2$

Combine two linear partial solutions.

$a_n = \alpha_1r_1^n + \alpha_2r_2^n + \ldots + \alpha_kr_k^n + \ldots$

$a_n = -n - 3/2 + \alpha_13^n$

Use initial conditions to solve.

$$a_0 = 1/3 = -0 - 3/2 + \alpha_13^0 = -3/2 + \alpha_1$$

$\alpha_1, \alpha_2, \ldots, \alpha_k$

$\alpha_1 = 11/6$

$a_n = -n - 3/2 + (11/6)3^n$

Dealing With Other Recurrence Forms

$$T(n) = 1 \qquad n \le 2$$
$$T(n) = T(\lg n) + n \qquad n > 2$$

Change variables:

Consider $n = \left.2^{2^{\cdots^{2}}}\right\}m$

$$T\left(\left.2^{2^{\cdots^{2}}}\right\}m\right) = 1 \qquad m=0$$

$$T\left(\left.2^{2^{\cdots^{2}}}\right\}m\right) = T\left(\left.2^{2^{\cdots^{2}}}\right\}m-1\right) + \left.2^{2^{\cdots^{2}}}\right\}m \qquad m>0$$

$$T'(m) = 1 \qquad m=0$$
$$T'(m) = T'(m-1) + \left.2^{2^{\cdots^{2}}}\right\}m \qquad m>0$$

Solve for T'(), then reverse change of variables for T().