Making a good guess

- No general way to guess the correct solutions to recurrences.
- Guessing a solution takes experience and, occasionally, creativity.

ex. $T(n) = 2T(\lfloor n/2 \rfloor + \underline{17}) + n$

We also guess $T(n) = O(n \lg n)$

- Making guess provides loose upper bound and lower bound. Then improve the gap.

Making a Good Guess

- If a recurrence is similar to one seen before, then guess a similar solution.
  - $T(n) = 3T(\lfloor n/3 \rfloor + 5) + n$ (Similar to $T(n) = 3T(\lfloor n/3 \rfloor) + n$)
    - When n is large, the difference between $n/3$ and $(n/3 + 5)$ is insignificant.
    - Hence, can guess $O(n \lg n)$.
- Method 2: Prove loose upper and lower bounds on the recurrence and then reduce the range of uncertainty.
  - E.g., start with $T(n) = \Omega(n)$ & $T(n) = O(n^2)$.
  - Then lower the upper bound and raise the lower bound.

Avoiding Pitfalls

- Be careful not to misuse asymptotic notation. For example:
  - We can falsely prove $T(n) = O(n)$ by guessing $T(n) \le cn$ for $T(n) = 2T(\lfloor n/2 \rfloor) + n$

$$T(n) \le 2c\lfloor n/2 \rfloor + n$$
$$\le cn + n$$
$$= O(n) \Leftarrow \text{Wrong!}$$

  - We are supposed to prove that $T(n) \le c\,n$ for all $n>N$, according to the definition of $O(n)$.
- Remember: prove the *exact form* of inductive hypothesis.

Changing Variables

- Use algebraic manipulation to turn an unknown recurrence into one similar to what you have seen before.
  - Example: $T(n) = 2T(n^{1/2}) + \lg n$
  - Rename **$m = \lg n$** and we have
    $T(2^m) = 2T(2^{m/2}) + m$
  - Set $S(m) = T(2^m)$ and we have
    $S(m) = 2S(m/2) + m \Rightarrow S(m) = O(m \lg m)$
  - Changing back from $S(m)$ to $T(n)$, we have
    $T(n) = T(2^m) = S(m) = O(m \lg m) = O(\lg n \lg \lg n)$

Substitution Method

- **Guess** the form of the solution, then **use mathematical induction** to show it correct.
  - Substitute guessed answer for the function when the inductive hypothesis is applied to smaller values – hence, the name.
- Works well when the solution is easy to guess.
- No general way to guess the correct solution.

Solving Recurrences

- The substitution method (CLR 4.3)
  - A.k.a. the "making a good guess method"
  - Guess the form of the answer, then use induction to find the constants and show that solution works
  - Examples:
    - $T(n) = 2T(n/2) + \Theta(n) \rightarrow T(n) = \Theta(n \lg n)$
    - $T(n) = 2T(\lfloor n/2 \rfloor) + n \rightarrow$???

Solving Recurrences

- The substitution method (CLR 4.3)
  - A.k.a. the "making a good guess method"
  - Guess the form of the answer, then use induction to find the constants and show that solution works
  - Examples:
    - $T(n) = 2T(n/2) + \Theta(n) \rightarrow T(n) = \Theta(n \lg n)$
    - $T(n) = 2T(\lfloor n/2 \rfloor) + n \rightarrow T(n) = \Theta(n \lg n)$
    - $T(n) = 2T(\lfloor n/2 \rfloor) + 17) + n \rightarrow$???

Solving Recurrences

- The substitution method (CLR 4.3)
  - A.k.a. the "making a good guess method"
  - Guess the form of the answer, then use induction to find the constants and show that solution works
  - Examples:
    - $T(n) = 2T(n/2) + \Theta(n) \rightarrow T(n) = \Theta(n \lg n)$
    - $T(n) = 2T(\lfloor n/2 \rfloor) + n \rightarrow T(n) = \Theta(n \lg n)$
    - $T(n) = 2T(\lfloor n/2 \rfloor + 17) + n \rightarrow \Theta(n \lg n)$

Example – Exact Function

Recurrence: $T(n) = 1$ if $n = 1$

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

- ***Guess:*** $T(n) = n \lg n + n$.
- ***Induction:***
  - ***Basis:*** $n = 1 \Rightarrow n \lg n + n = 1 = T(n)$.
  - ***Hypothesis:*** $T(k) = k \lg k + k$ for all $k < n$.
  - ***Inductive Step:*** $T(n) = 2\,T(n/2) + n$

$$\begin{aligned} &= 2\,((n/2)\lg(n/2) + (n/2)) + n \\ &= n\,(\lg(n/2)) + 2n \\ &= n \lg n - n + 2n \\ &= n \lg n + n \end{aligned}$$

Example – With Asymptotics

To Solve: $T(n) = 3T(\lfloor n/3 \rfloor) + n$

- Guess: $T(n) = O(n \lg n)$
- Need to prove: $T(n) \le cn \lg n$, for some $c > 0$.
- Hypothesis: $T(k) \le ck \lg k$, for all $k < n$.
- Calculate:

$$\begin{aligned} T(n) &\le 3c\lfloor n/3 \rfloor \lg \lfloor n/3 \rfloor + n \\ &\le c\,n \lg (n/3) + n \\ &= c\,n \lg n - c\,n \lg 3 + n \\ &= c\,n \lg n - n\,(c \lg 3 - 1) \\ &\le c\,n \lg n \end{aligned}$$

(The last step is true for $c \ge 1 / \lg 3$.)

Example – With Asymptotics

To Solve: $T(n) = 3T(\lfloor n/3 \rfloor) + n$

- To show $T(n) = \Theta(n \lg n)$, must show both upper and lower bounds, i.e., $T(n) = O(n \lg n)$ **AND** $T(n) = \Omega(n \lg n)$
- (Can you find the mistake in this derivation?)
- Show: $T(n) = \Omega(n \lg n)$
- Calculate:

$$\begin{aligned} T(n) &\ge 3c\lfloor n/3 \rfloor \lg \lfloor n/3 \rfloor + n \\ &\ge c\,n \lg (n/3) + n \\ &= c\,n \lg n - c\,n \lg 3 + n \\ &= c\,n \lg n - n\,(c \lg 3 - 1) \\ &\ge c\,n \lg n \end{aligned}$$

(The last step is true for $c \le 1 / \lg 3$.)

Example – With Asymptotics

If $T(n) = 3T(\lfloor n/3 \rfloor) + O(n)$, as opposed to $T(n) = 3T(\lfloor n/3 \rfloor) + n$, then rewrite $T(n) \le 3T(\lfloor n/3 \rfloor) + cn$, $c > 0$.

- To show $T(n) = O(n \lg n)$, use second constant ***d***, different from ***c***.
- Calculate:

$$\begin{aligned} T(n) &\le 3d\lfloor n/3 \rfloor \lg \lfloor n/3 \rfloor + c\,n \\ &\le d\,n \lg (n/3) + cn \\ &= d\,n \lg n - d\,n \lg 3 + cn \\ &= d\,n \lg n - n\,(d \lg 3 - c) \\ &\le d\,n \lg n \end{aligned}$$

(The last step is true for $d \ge c / \lg 3$.)

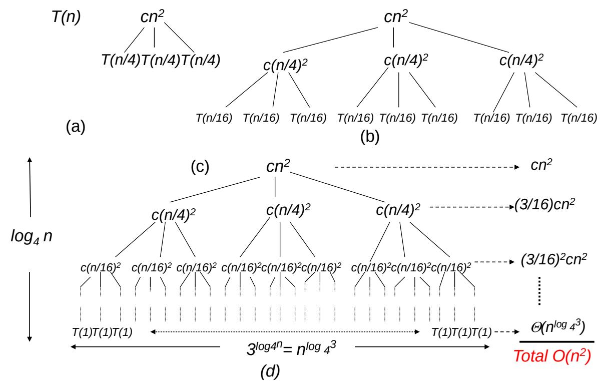
It is <u>OK</u> for d to depend on c.

4.4 The Recursion-tree Method

- Idea:
  - Each node represents the cost of a single subproblem.
  - Sum up the costs with each level to get level cost.
  - Sum up all the level costs to get total cost.
- Particularly suitable for divide-and-conquer recurrence.
- Best used to generate a good guess, tolerating "sloppiness".
- If trying carefully to draw the recursion-tree and compute cost, then used as direct proof.

L4.14

Recursion Tree for $T(n)=3T(n/4)+cn^2$



Solution to $T(n)=3T(n/4)+cn^2$)

- The height is $\log_4 n$,
- #leaf nodes = $3^{\log_4 n} = n^{\log_4 3}$. Leaf node cost: $T(1)$.
- Total cost $T(n)=cn^2+(3/16)\,cn^2+(3/16)^2\,cn^2+ \cdots +(3/16)^{\log_4 n-1}\,cn^2+\Theta(n^{\log_4 3})$
  $=(1+3/16+(3/16)^2+\cdots+(3/16)^{\log_4 n-1})\,cn^2+\Theta(n^{\log_4 3})$
  $<(1+3/16+(3/16)^2+\cdots+(3/16)^m+\cdots)\,cn^2+\Theta(n^{\log_4 3})$
  $=(1/(1-3/16))$ $cn^2+\Theta(n^{\log_4 3})$ Equation A.6 (pp. 1147)
  $=16/13cn^2+\Theta(n^{\log_4 3})$
  $=O(n^2)$.

$$\sum_{k=0}^{\infty} x^k = 1/(1-x)$$

L4.16

Solution to $T(n)=3T(n/4)+cn^2$) Using Summation

$$T(n)=\sum_{i=0}^{\log_4 n-1}\left(\frac{3}{16}\right)^i cn^2+\Theta\left(n^{\log_4 3}\right)$$
$$<\sum_{i=0}^{\infty}\left(\frac{3}{16}\right)^i cn^2+\Theta\left(n^{\log_4 3}\right)$$
$$=\frac{1}{1-(3/16)}cn^2+\Theta\left(n^{\log_4 3}\right)$$
$$=\frac{16}{13}cn^2+\Theta(n^{\log_4 3})$$
$$=O(n^2)$$

L4.17

Prove the above Guess by substitution method

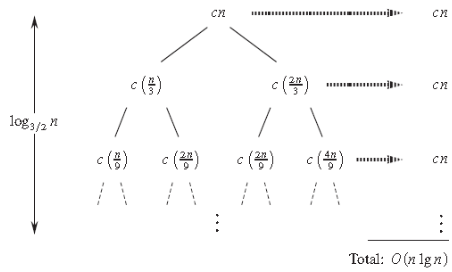
We want to show that $T(n) \le dn^2$ for some constant $d > 0$.
using the same constant $c > 0$ as before, we have

$$
\begin{aligned}
T(n) &\le 3T(\lfloor n/4 \rfloor) + cn^2 \\
&\le 3d\lfloor n/4 \rfloor^2 + cn^2 \\
&\le 3d(n/4)^2 + cn^2 \\
&= \frac{3}{16} dn^2 + cn^2 \\
&\le dn^2,
\end{aligned}
$$

Where the last step holds as long as $d \ge (16/13)c$.

Another example

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



- The longest path from root to leaf is n → (2/3)n → $(2/3)^2$ n → … → 1
- Since $(2/3)^k n = 1$ when $k = \log_{3/2} n$, the height of tree is $\log_{3/2} n$.

Verify by substitution method

$$
\begin{aligned}
T(n) &\le T(n/3) + T(2n/3) + cn \\
&\le d(n/3)\lg(n/3) + d(2n/3)\lg(2n/3) + cn \\
&= (d(n/3)\lg n - d(n/3)\lg 3) + (d(2n/3)\lg n - d(2n/3)\lg(3/2)) + cn \\
&= dn\lg n - d((n/3)\lg 3 + (2n/3)\lg(3/2)) + cn \\
&= dn\lg n - d((n/3)\lg 3 + (2n/3)\lg 3 - (2n/3)\lg 2 + cn \\
&= dn\lg n - dn(\lg 3 - 2/3) + cn \\
&\le dn\lg n,
\end{aligned}
$$

As long as $d \ge c/lg3 – (2/3))$.