Analysis of Algorithms
CS 477/677

Instructor: Monica Nicolescu

Lecture 5

Methods for Solving Recurrences

- *Iteration method*
- *Substitution method*
- Recursion tree method
- Master method

The recursion-tree method

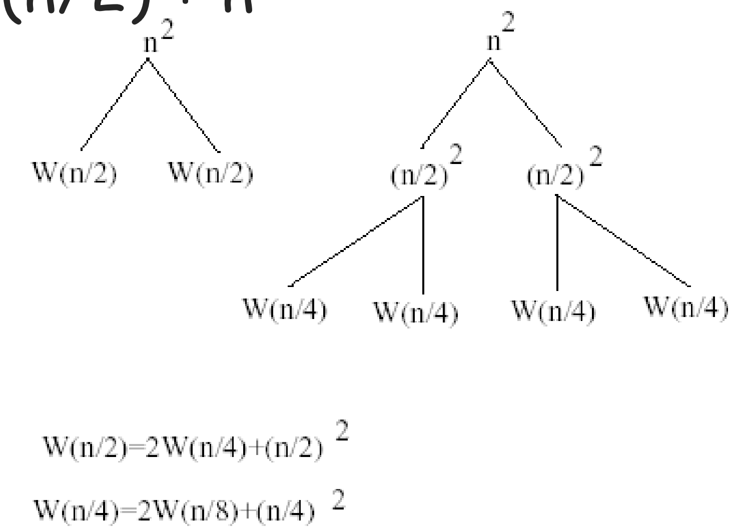
Convert the recurrence into a tree:

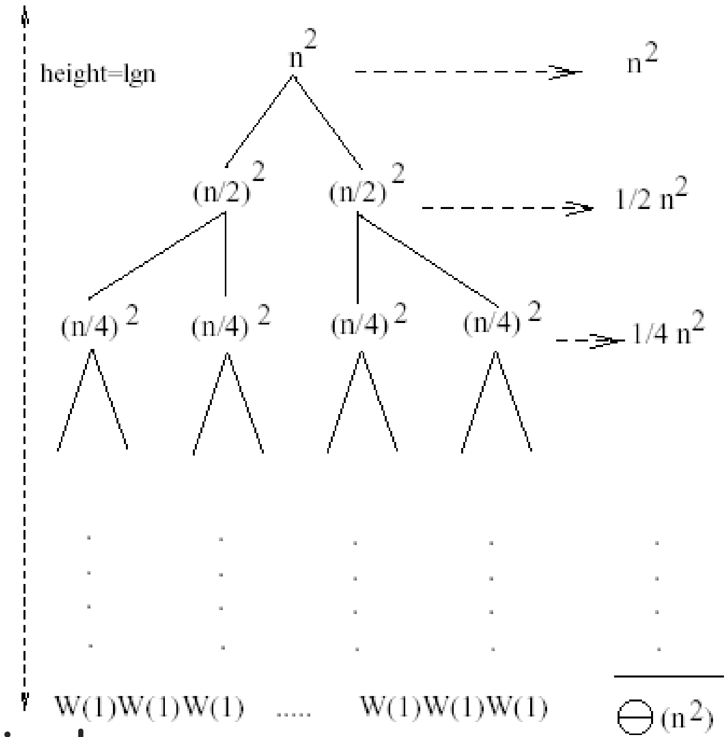
- Each node represents the cost incurred at that level of recursion
- Sum up the costs of all levels

Used to "guess" a solution for the recurrence

Example 1

$W(n) = 2W(n/2) + n^2$





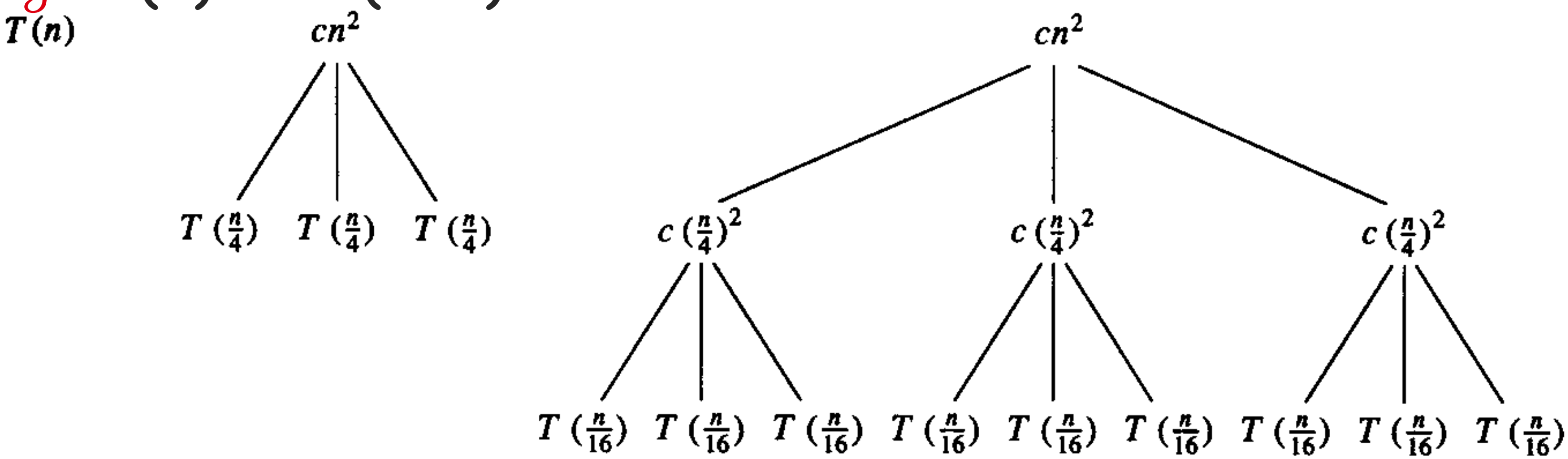
- Subproblem size at level i is: $n/2^i$
- Subproblem size hits 1 when $1 = n/2^i \Rightarrow i = lgn$
- Cost of the problem at level i = $(n/2^i)^2$ No. of nodes at level $i = 2^i$
- Total cost:

$$W(n) = \sum_{i=0}^{lgn-1} \frac{n^2}{2^i} + 2^{lgn} W(1) = n^2 \sum_{i=0}^{lgn-1} \left(\frac{1}{2}\right)^i + n \le n^2 \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i + O(n) \le \frac{n^2}{1-\frac{1}{2}} + O(n)$$

$\Rightarrow W(n) = O(n^2)$

Example 2

E.g.: $T(n) = 3T(n/4) + cn^2$



- Subproblem size at level i is: $n/4^i$
- Subproblem size hits 1 when $1 = n/4^i \Rightarrow i = \log_4 n$
- Cost of a node at level i = $c(n/4^i)^2$
- Number of nodes at level i = $3^i \Rightarrow$ last level has $3^{\log_4 n} = n^{\log_4 3}$ nodes
- Total cost:

+

$\Rightarrow T(n) = O(n^2)$

Example 2 - Substitution

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

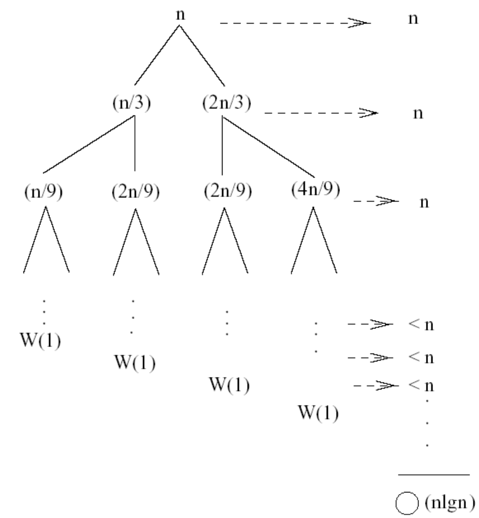
- Guess: $T(n) = O(n^2)$
  - Induction goal: $T(n) \le dn^2$, for some d and $n \ge n_0$
  - Induction hypothesis: $T(n/4) \le d\,(n/4)^2$
- Proof of induction goal:

$$\begin{aligned}
T(n) &= 3T(n/4) + cn^2 \\
&\le 3d\,(n/4)^2 + cn^2 \\
&= (3/16)\,d\,n^2 + cn^2 \\
&= d\,n^2 + cn^2 + (3/16)\,d\,n^2 - d\,n^2 \\
&= d\,n^2 + cn^2 - (13/16)\,d\,n^2 \\
&\le d\,n^2 \qquad \text{if: } cn^2 - (13/16)\,d\,n^2 \le 0 \\
&\qquad\qquad\quad d \ge (16/13)c
\end{aligned}$$

Therefore: $T(n) = O(n^2)$

Example 3

$W(n) = W(n/3) + W(2n/3) + n$



- The longest path from the root to a leaf is:

  $n \rightarrow (2/3)n \rightarrow (2/3)^2\, n \rightarrow \ldots \rightarrow 1$

- Subproblem size hits 1 when

  $1 = (2/3)^i n \Leftrightarrow i = \log_{3/2} n$

- Cost of the problem at level i = n

- Total cost:

$$\underbrace{W(n) \le n + n +}_{\text{for all levels}} \ldots = n * \sum_{i=0}^{\log_{3/2} n} 1 = n \frac{\lg n}{\lg 3/2} = \frac{1}{\lg 3/2} n \lg n$$

$\Rightarrow W(n) = O(n\lg n)$

Example 3 - Substitution

$$W(n) = W(n/3) + W(2n/3) + n$$

- Guess: $W(n) = O(nlgn)$
  - Induction goal: $W(n) \le dnlgn$, for some d and $n \ge n_0$
  - Induction hypothesis: $W(k) \le d\,klgk$ for any $k < n$ $(n/3, 2n/3)$
- Proof of induction goal:

Try it out as an exercise!!

Master method

- "Cookbook" for solving recurrences of the form:

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

where, $a > 0$, $b > 1$, and $f(n) > 0$

Idea: compare $f(n)$ with $n^{\log_b a}$

- $f(n)$ is asymptotically smaller or larger than $n^{\log_b a}$ by a polynomial factor n^{ε}
- $f(n)$ is asymptotically equal with $n^{\log_b a} \lg^k n$ $(k \geq 0)$

Master method

- "Cookbook" for solving recurrences of the form:

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

where, $a > 0$, $b > 1$, and $f(n) > 0$

Case 1: if $f(n) = O(n^{\log_b a - \varepsilon})$ for some $\varepsilon > 0$, then: $T(n) = \Theta(n^{\log_b a})$

Case 2: if $f(n) = \Theta(n^{\log_b a} \lg^k n)$, (for some $k \geq 0$) then:

$$T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$$

Case 3: if $f(n) = \Omega(n^{\log_b a + \varepsilon})$ for some $\varepsilon > 0$, and if

$af(n/b) \leq cf(n)$ for some $c < 1$ and all sufficiently large n, then:

regularity condition

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

Examples

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

$a = 2$, $b = 2$, $\log_2 2 = 1$

Compare $f(n) = n$ with $n^{\log_2 2} = n$

$\Rightarrow f(n) = \Theta(n \lg^0 n) \Rightarrow$ Case 2

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

Examples (cont.)

$$T(n) = 9T(n/3) + n$$

$a = 9$, $b = 3$, $\log_3 9 = 2$

Compare $f(n) = n$ with $n^{\log_3 9} = n^2$

$\Rightarrow f(n) = O(n^{2-\varepsilon})$ for $\varepsilon \leq 1$ Case 1

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

Examples

$$T(n) = 2T(n/2) + n^2$$

$a = 2$, $b = 2$, $\log_2 2 = 1$

Compare $f(n) = n^2$ with n

$\Rightarrow f(n) = \Omega(n^{1+\varepsilon})$

Case 3 $\Rightarrow$ verify regularity cond.: $a\, f(n/b) \le c\, f(n)$

$\Rightarrow 2\, n^2/4 \le c\, n^2 \Rightarrow c = \frac{1}{2}$ is a solution $(c<1)$

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

Examples (cont.)

$$T(n) = 2T(n/2) + \sqrt{n}$$

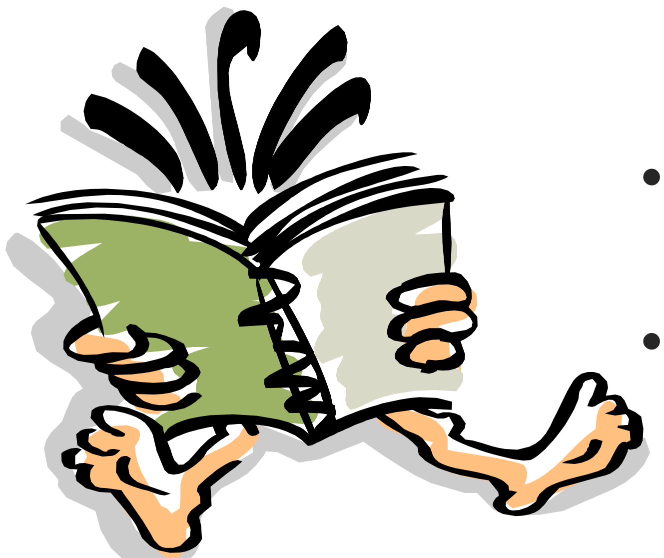
$a = 2$, $b = 2$, $\log_2 2 = 1$

Compare $f(n) = n^{1/2}$ with n

$\Rightarrow f(n) = O(n^{1-\varepsilon})$ for $\varepsilon \le 1/2$ Case 1

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

Readings



- For this lecture
  - Section 4.4
- Coming next
  - Sections 4.5, 2.1, 2.2