Analysis of Algorithms
CS 477/677

Instructor: Monica Nicolescu

Lecture 4

Methods for Solving Recurrences

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

The Iteration Method

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

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

$\quad = c + c + T(n/4)$

$\quad = c + c + c + T(n/8)$

$T(n/2) = c + T(n/4)$

$T(n/4) = c + T(n/8)$

Assume $n = 2^k$

$$T(n) = \underbrace{c + c + \ldots + c}_{k \text{ times}} + T(1)$$

$$= c\lg n + T(1)$$

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

Iteration Method – Example

$$T(n) = n + 2T(n/2) \qquad \text{Assume: } n = 2^k$$

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

$$\begin{aligned}
T(n) &= n + 2T(n/2) \\
&= n + 2(n/2 + 2T(n/4)) \\
&= n + n + 4T(n/4) \\
&= n + n + 4(n/4 + 2T(n/8)) \\
&= n + n + n + 8T(n/8) \\
\ldots &= in + 2^iT(n/2^i) \\
&= kn + 2^kT(1) \\
&= n\lg n + nT(1) = \Theta(n\lg n)
\end{aligned}$$

Iteration Method – Example

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

$$\begin{aligned}
T(n) &= n + T(n-1) \\
&= n + (n-1) + T(n-2) \\
&= n + (n-1) + (n-2) + T(n-3) \\
\ldots \quad &= n + (n-1) + (n-2) + \ldots + 2 + T(1) \\
&= n(n+1)/2 - 1 + T(1) \\
&= n^2 + T(1) = \Theta(n^2)
\end{aligned}$$

The substitution method

1. Guess a solution

2. Use induction to prove that the solution works

Substitution method

- Guess a solution
  - $T(n) = O(g(n))$
  - Induction goal: apply the definition of the asymptotic notation
    - $T(n) \le d\ g(n)$, for some $d > 0$ and $n \ge n_0$
  - Induction hypothesis: $T(k) \le d\ g(k)$ for all $k < n$
- Prove the induction goal
  - Use the **induction hypothesis** to find some values of the constants d and n_0 for which the **induction goal** holds

Example: Binary Search

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

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

$$T(n) = T(n/2) + c \le d \lg(n/2) + c$$

$$= d \lg n - d + c \le d \lg n$$

$$\text{if: } -d + c \le 0,\ d \ge c$$

Example: Binary Search

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

- Boundary conditions:
  - Base case: $n_0 = 1 \Rightarrow T(1) = c$ has to verify condition:

    $T(1) \le d \lg 1 \Rightarrow c \le d \lg 1 = 0$ – contradiction
  - Choose $n_0 = 2 \Rightarrow T(2) = 2c$ has to verify condition:

    $T(2) \le d \lg 2 \Rightarrow 2c \le d \lg 2 = d \Rightarrow$ choose $d \ge 2c$
- We can similarly prove that $T(n) = \Omega(\lg n)$ and therefore: $T(n) = \Theta(\lg n)$

Example 2

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

- Guess: $T(n) = O(n^2)$
  - Induction goal: $T(n) \le c\, n^2$, for some c and $n \ge n_0$
  - Induction hypothesis: $T(n-1) \le c(n-1)^2$
- Proof of induction goal:

$$T(n) = T(n-1) + n \le c\,(n-1)^2 + n$$

$$= cn^2 - (2cn - c - n) \le cn^2$$

$$\text{if: } 2cn - c - n \ge 0 \Rightarrow c \ge n/(2n-1) \Rightarrow c \ge 1/(2 - 1/n)$$

  - For $n \ge 1 \Rightarrow 2 - 1/n \ge 1 \Rightarrow$ any $c \ge 1$ will work

Example 2

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

- Boundary conditions:
  - Base case: $n_0 = 1 \Rightarrow T(1) = 1$ has to verify condition:

    $T(1) \leq c\,(1)^2 \Rightarrow 1 \leq c \Rightarrow$ OK!
- We can similarly prove that $T(n) = \Omega(n^2)$ and therefore: $T(n) = \Theta(n^2)$

Example 3

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

- Guess: $T(n) = O(n\lg n)$
  - Induction goal: $T(n) \le cn \lg n$, for some c and $n \ge n_0$
  - Induction hypothesis: $T(n/2) \le cn/2 \lg(n/2)$
- Proof of induction goal:

$$T(n) = 2T(n/2) + n \le 2c\,(n/2)\lg(n/2) + n$$

$$= cn \lg n - cn + n \le cn \lg n$$

$$\text{if: } -cn + n \le 0 \Rightarrow c \ge 1$$

Example 3

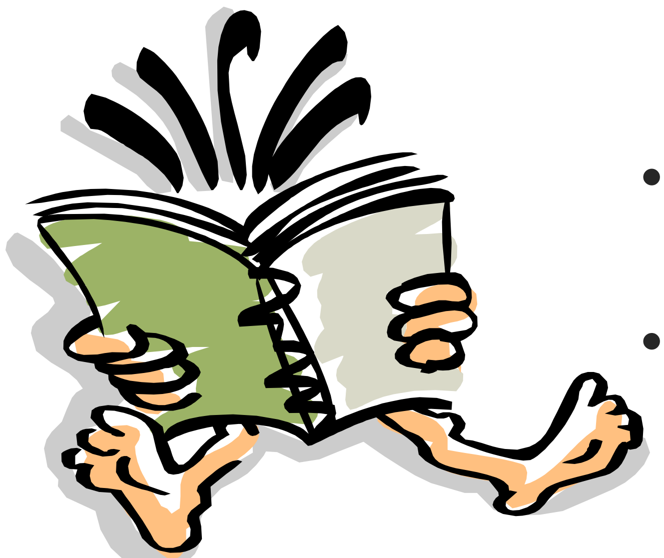
$$T(n) = 2T(n/2) + n$$

- Boundary conditions:
  - Base case: $n_0 = 1 \Rightarrow T(1) = 1$ has to verify condition:

    $T(1) \le cn_0 lgn_0 \Rightarrow 1 \le c * 1 * lg1 = 0$ – contradiction
  - Choose $n_0 = 2 \Rightarrow T(2) = 4$ has to verify condition:

    $T(2) \le c * 2 * lg2 \Rightarrow 4 \le 2c \Rightarrow$ choose $c = 2$
- We can similarly prove that $T(n) = \mathbf{\Omega}(nlgn)$ and therefore: $T(n) = \mathbf{\Omega}(nlgn)$

Readings



- For this lecture
  - Section 4.3
- Coming next
  - Sections 4.4, 4.5