CompSci 230
Discrete Math for Computer Science
Recurrence Relations

December 3, 2013
Prof. Rodger

Announcements
• Reading this week Chap. 8.1-8.3, Chap10.1-10.4, 10.7-10.8, Chapter 11.1
  – Looks like a lot, but also a lot of review from CompSci 201, plus some new items
• Last Recitation on Friday (optional)
• Test back today

Recurrence Relations
- model lots of problems
• The Tower of Hanoi
• Divide and conquer algorithms
  – Sorting algorithm mergesort
  – Sorting algorithm quicksort
• Tree algorithms
  – Searching for an element in a binary search tree
  – Listing out all elements in a binary search tree

Solving a recurrence relation
• Problem sets up as a recurrence
  – Must have a base case
• Solve the recurrence
  – Use substitution
• Prove correctness
  – Proof by induction
Example 1

• $a_n = a_{n-1} + c$
• $a_0 = 1$

• Solve recurrence
• Then prove true by induction
• What is this an example of?

Proof by induction

Basis: $a_0 = 1$

$a_n = \ldots$

Assume: for all $k < n$

$a_n = a_{n-1} + c$ Show true for $k = n$

$\ldots$

Worst case binary search tree
Example 2

- \( a_n = 2a_{n-1} + c \)
- \( a_0 = 0 \)

Solve recurrence
Then prove true by induction
What is this an example of?

Example 2

Proof by Induction

Basis: \( a_0 = 0 \)

Assume: \( a_k = \) for all \( k < n \)

\[ a_n = 2a_{n-1} + c \]

\[ = c(2^n - 1) \]
Towers of Hanoi

- Figures
  - Figs 1-4
    - problem size n-1
  - Figs 4-5
    - Constant work
  - Figs 5-7
    - problem size n-1

Example 3

- $a_n = 2a_{n/2} + c$
- $a_1 = c$

- Solve recurrence
- Then prove true by induction
- What is this an example of?

Example 3

\[
\begin{align*}
a_1 &= c \\
a_n &= 2a_{n/2} + c \\
   &= 2(2a_{n/4} + c) + c \\
   &= 2^2 a_{n/4} + 2c + c \\
   &= 2^2 (2a_{n/8} + c) + 2c + c \\
   &= 2^3 a_{n/8} + 4c + 2c + c \\
   &\vdots \\
   &= 2^k a_{n/2^k} + c(2^k - 1) \\
   &= n * a_1 + c(n - 1) \\
   &= n * c + cn - c \\
   &= 2cn - c
\end{align*}
\]
Prove by induction
Basis: \( a_1 = c \)
\[ a_n = 2cn - c \]
\[ a_1 = \]
Assume: \( a_k = 2c^k - c \) for \( k < n \)
\[ a_n = 2a_n/2 + c \]
\[ = \]
\[ = \]
\[ = \]

Traversal in binary search tree
preorder, postorder, inorder

Example 4
- \( a_n = 2a_n/2 + cn \)
- \( a_1 = c \)
- Solve recurrence
- Then prove true by induction
- What is this an example of?

Example 4
\[ a_1 = c \]
\[ a_n = 2a_n/2 + cn \]
\[ = \]
\[ = \]
\[ = \]
\[ = \]
\[ = \]
\[ \ldots \]
\[ \rightarrow \]
\[ k = \]
Example 4

- $n \times a_1 + (\log_2 n)c_n$
- $n^c + c_n \log_2 n$
- $c_n(1 + \log_2 n)$

Proof by induction

Basis: $a_1 = c$

Assume: for all $k < n$

$\begin{align*}
  a_n &= 2 \times \frac{a_n}{2} + c_n \\
  &= 2 \times \left(\frac{c_n}{2} + c_n \log_2 \frac{n}{2}\right) + c_n \\
  &= c_n + c_n (\log_2 n + \log_2 \frac{1}{2}) + c_n \\
  &= c_n(2 + \log_2 n - 1) = c_n(1 + \log_2 n)
\end{align*}$

$cn(1 + \log_2 n)$

$c(1)(1 + \log_2 1)$ for $n=1$

$c \times (1 + 0) = c$

$ak = ck + ck \log_2 k$

MergeSort

- $n \log n$

Definition

- A linear homogeneous recurrence relation of degree $k$ with constant coefficients is a recurrence relation of the form

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

- Where $c_i$ are real numbers and $c_k \neq 0$
8.2 - Theorem 1

Let $c_1$ and $c_2$ be real numbers. Suppose that $r^2 - c_1 r - c_2 = 0$ has two distinct roots $r_1$ and $r_2$. Then the sequence $\{a_n\}$ is a solution of the recurrence

$$a_n = c_1 a_{n-1} + c_2 a_{n-2}$$

if and only if

$$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$$

for all $n$ where $\alpha_1$ and $\alpha_2$ are constants.

8.2 - Theorem 2

- Let $c_1$ and $c_2$ be real numbers with $c_2 \neq 0$. Suppose that $r^2 - c_1 r - c_2 = 0$ has only one root $r_0$. A sequence $\{a_n\}$ is a solution of the recurrence relation

$$a_n = c_1 a_{n-1} + c_2 a_{n-2}$$

if and only if

$$a_n = \alpha_1 r_0^n + \alpha_2 n r_0^n$$

for all $n$ where $\alpha_1$ and $\alpha_2$ are constants.

Example

- What is the solution to the recurrence relation $a_n = a_{n-1} + 2 a_{n-2}$ with $a_0 = 2, a_1 = 7$?

Suppose that $r^2 - c_1 r - c_2 = 0$ has two distinct roots $r_1$ and $r_2$. Then the sequence $\{a_n\}$ is a solution of the recurrence

$$a_n = c_1 a_{n-1} + c_2 a_{n-2}$$

if and only if

$$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$$

for all $n$ where $\alpha_1$ and $\alpha_2$ are constants.

Many other theorems

- See theorems 2-6 in Chapter 8.2
Theorem 6

Suppose that \( \{a_n\} \) satisfies the linear nonhomogeneous recurrence relation

\[
a_n = c_1a_{n-1} + c_2a_{n-2} + \cdots + c_k a_{n-k} + F(n)
\]

where \( c_1, c_2, \ldots, c_k \) are real numbers, and

\[
F(n) = (b_t n^t + b_{t-1} n^{t-1} + \cdots + b_1 n + b_0)s^n,
\]

where \( b_0, b_1, \ldots, b_t \) and \( s \) are real numbers. When \( s \) is not a root of the characteristic equation of the associated linear homogeneous recurrence relation, there is a particular solution of the form

\[
(p_t n^t + p_{t-1} n^{t-1} + \cdots + p_1 n + p_0)s^n
\]

Theorem 6 (cont)

When \( s \) is a root of this characteristic equation and its multiplicity is \( m \), there is a particular solution of the form

\[
n^m(p_t n^t + p_{t-1} n^{t-1} + \cdots + p_1 n + p_0)s^n
\]

Theorem 1 in 8.3

**Theorem 1**: Let \( f \) be an increasing function that satisfies the recurrence relation

\[
f(n) = af(n/b) + c
\]

whenever \( n \) is divisible by \( b \), where \( a \geq 1 \), \( b \) is an integer greater than 1, and \( c \) is a positive real number. Then

\[
f(n) \text{ is } \begin{cases} O(n^{\log_b a}) & \text{if } a > 1 \\ O(\log n) & \text{if } a = 1 \end{cases}
\]

Furthermore, when \( n = b^k \) and \( a \neq 1 \), where \( k \) is a positive integer,

\[
f(n) = C_1 n^{\log_b a} + C_2
\]

where \( C_1 = f(1) + c/(a-1) \) and \( C_2 = -c/(a-1) \).

Master Theorem in 8.3

**Theorem 2. Master Theorem**: Let \( f \) be an increasing function that satisfies the recurrence relation

\[
f(n) = af(n/b) + cn^d
\]

whenever \( n = b^k \), where \( k \) is a positive integer greater than 1, and \( c \) and \( d \) are real numbers with \( c \) positive and \( d \) nonnegative. Then

\[
f(n) \text{ is } \begin{cases} O(n^d) & \text{if } a < b^d, \\ O(n^d \log n) & \text{if } a = b^d, \\ O(n^{\log_b a}) & \text{if } a > b^d. \end{cases}
\]