Last Class: Solution to Important Recurrence Types

One (constant) operation reduces problem size by one.

$T(n) = T(n-1) + c$ *for n>1* $T(1) = d$

Solution: $T(n) = (n-1)c + d$ *linear, e.g., factorial*

A pass through input reduces problem size by one.

$T(n) = T(n-1) + cn$ *for n>1* $T(1) = d$

Solution: $T(n) = [n(n+1)/2 - 1]\, c + d$ *quadratic, e.g., insertion sort*

One (constant) operation reduces problem size by half.

$T(n) = T(n/2) + c$ *for n>1* $T(1) = d$

Solution: $T(n) = c \log_2 n + d$ *logarithmic, e.g., binary search*

Note: you can have similar solution with an arbitrary base b

A pass through input reduces problem size by half.

$T(n) = 2T(n/2) + cn$ *for n>1* $T(1) = d$

Solution: $T(n) = cn \log_2 n + d\, n$ *$n \log_2 n$, e.g., mergesort*

Last Class: Linear second-order recurrences with constant coefficients

$$ax(n) + bx(n-1) + cx(n-2) = f(n) \qquad a \neq 0$$

Second-order term — A function of n

a, b, and c are constant coefficients.

$f(n) = 0$ **homogeneous**

$f(n) \neq 0$ **inhomogeneous**

Last Class: Linear second-order recurrences with constant coefficients - Homogeneous case

Homogeneous case:

$$ax(n)+bx(n-1)+cx(n-2)=0 \qquad a\neq 0$$

Characteristic equation:

$$ar^2+br+c=0$$

Roots of the characteristic equation determine the general solution:

$$case1 \qquad x(n)=\alpha r_1^n+\beta r_2^n \qquad r_1\neq r_2 \qquad r_1,r_2\in R$$

$$case2 \qquad x(n)=\alpha r^n+\beta n r^n$$

$$case3 \qquad x(n)=\gamma^n\left[\alpha\cos n\theta+\beta\sin n\theta\right]$$

$$r_{1,2}=u\pm jv \qquad \gamma=\sqrt{u^2+v^2} \qquad \theta=\arctan\frac{v}{u}$$

Last Class: Linear second-order recurrences with constant coefficients – Inhomogeneous Case

Inhomogeneous case:

$$ax(n)+bx(n-1)+cx(n-2)=f(n) \qquad a\neq 0$$

Its general solution is the summation of one of its particular solution and the general solution of

$$ax(n)+bx(n-1)+cx(n-2)=0$$

- Nontrivial problem for an arbitrary $f(n)$
- Can be solved for special $f(n)$, e.g., a constant

Example

$$x(n) - 10x(n-1) + 25x(n-2) = 16$$

The homogeneous case: $x(n) - 10x(n-1) + 25x(n-2) = 0$

Step 1: find a particular solution of the inhomogeneous function

Assume $x(n) = c$ ⇒ $c = 1$

Step 2: find the general solution of the homogeneous function

$$x(n) = \alpha(5)^n + \beta n(5)^n$$

The general solution of inhomogeneous function

$$x(n) = \alpha(5)^n + \beta n(5)^n + 1$$

The particular solution can be obtained given the initial condition!

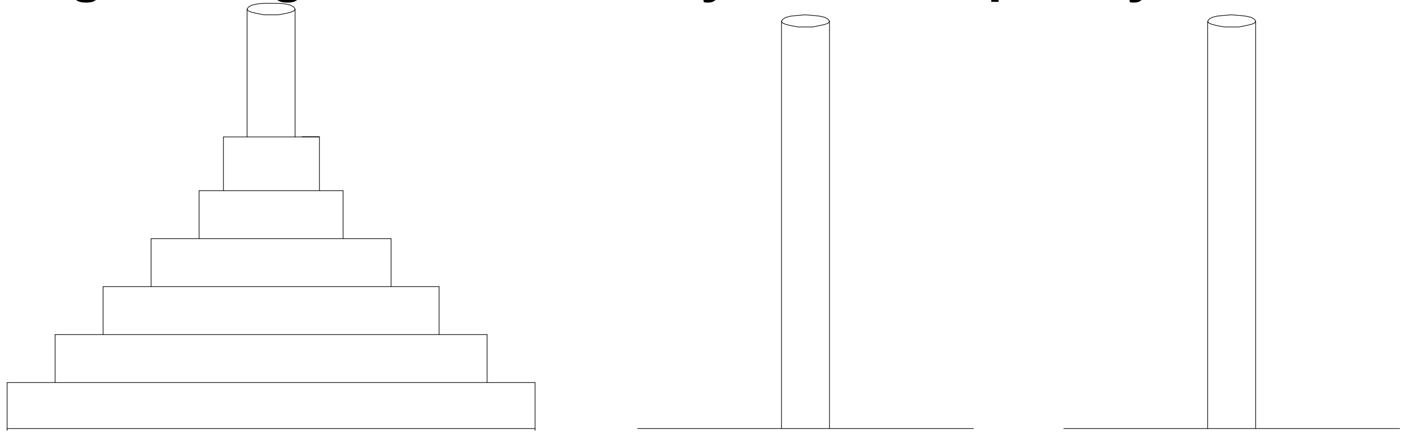
Applications of Linear 2nd Order Recurrences: Example: Tower of Hanoi

***n* different-size disks, 3 pegs, move disks from the left peg to the right one using the middle one as an auxiliary**

Rules:

- move one disk each time
- cannot place a larger disk on top of a smaller one

Design an algorithm and analyze its complexity



Algorithm Complexity

Let *M*(*n*) be the number of needed moves

Initialization *M*(1)=1

Recurrence $M(n) = M(n-1) + 1 + M(n-1)$ **for** $n > 1$

Move the n-1 disks to the middle peg

Move the n^{th} disk to the right peg

Move the n-1 disks to the right peg

Solve using backward substitution

$$M(n) = 2M(n-1) + 1 \quad \text{for } n > 1$$

$$= 2^n - 1$$

Algorithm Complexity – Solving with Linear Second Order

$$M(n) = 2M(n-1) + 1 \quad \text{for } n > 1$$

$$M(n) - 2M(n-1) = 1$$

Solve the recurrence relation using linear 2nd order inhomogenous case

Homogenous $M(n) - 2M(n-1) = 0$

Characteristic function: $r^2 - 2r = 0$ **Roots:** $r_1 = 0$ and $r_2 = 2$

$$case1 \quad x(n) = \alpha r_1^n + \beta r_2^n \quad r_1 \neq r_2 \quad r_1, r_2 \in R$$

$$M(n) = \alpha 0^n + \beta 2^n = \beta 2^n$$

Algorithm Complexity – Solving with Linear Second Order

$$M(n) = 2M(n-1) + 1 \quad \text{for } n > 1$$

Homogenous $M(n) - 2M(n-1) = 0$

General solution of Homogenous: $M(n) = \beta 2^n$

Assume $M(n) = c$ **is a particular solution** ⇒ $c = 2c + 1 \Rightarrow c = -1$

General solution: $M(n) = \beta 2^n - 1$

$$M(1) = \beta 2^1 - 1 = 1 \Rightarrow \beta = 1$$

⇒ **Particular solution:** $M(n) = 2^n - 1$

Summary: Methods for Solving Recurrence Relations

Forward substitutions

Backward substitutions

Linear 2nd order recurrences with constant coefficients

Following the solution to important recurrence type if appliable

Example: Find the Number of Binary Digits (Recursive Algorithm)

Find the Number of Binary Digits in the Binary Representation of a Positive Decimal Integer using a recursive algorithm

ALGORITHM $BinRec(n)$
// **Input : A positive decimal integer n**
// **Output : The number of binary digits**
// **in n's binary representation**
if $n = 1$ **return** 1
else return $BinRec(\lfloor n/2 \rfloor) + 1$

Recurrence

$$A(n) = A(\lfloor n/2 \rfloor) + 1, \quad \text{for } n > 1$$
$$A(1) = 0$$

However, $\lfloor n/2 \rfloor \neq n/2$ in general

Smooth Functions

- **Eventually nondecreasing function:**

  $f(n_1) \leq f(n_2)$, for $n_0 \leq n_1 < n_2$

  e.g., n, $\log n$, n^2, 2^n Is $\sin(n)$ eventually nondecreasing?

- **Smooth function:**

  $f(n)$ is eventually nondecreasing and $f(2n) \in \Theta(f(n))$

  - $f(n)$ cannot grow too fast, e.g., n, $\log n$, n^2
  - 2^n, $n!$ are not smooth functions

Properties of Smooth Functions

- If $f(n)$ is a smooth function, for any constant integer $b \geq 2$

  $f(bn) \in \Theta(f(n))$ **(See Appendix B for the proof)**

- **Smoothness rule:**

  $T(n)$ is eventually non decreasing and $f(n)$ is a smooth function

  If $T(n) \in \Theta(f(n))$ for $n = b^k, b \geq 2$

  $\text{then} \quad T(n) \in \Theta(f(n))$ for every n

  - Analogous results hold for big O and big Ω

  (See Appendix B for the proof)

Example: Find the Number of Binary Digits (Recursive Algorithm)

Find the Number of Binary Digits in the Binary Representation of a Positive Decimal Integer using a recursive algorithm

ALGORITHM $BinRec(n)$

// **Input : A positive decimal integer n**

// **Output : The number of binary digits**

// **in n's binary representation**

if $n = 1$ **return** 1

else return $BinRec(\lfloor n/2 \rfloor) + 1$

Recurrence

$$A(n) = A(\lfloor n/2 \rfloor) + 1, \quad \textbf{for } n > 1$$

$$A(1) = 0$$

Example: Find the Number of Binary Digits (Recursive Algorithm)

Recurrence

$$A(n) = A(\lfloor n/2 \rfloor) + 1, \quad \textbf{for } n > 1$$
$$A(1) = 0$$

A smooth function

Compare to

$$B(n) = B(n/2) + 1, \quad \text{for } n > 1$$
$$B(1) = 0$$

$$B(n) = \log_2 n \in \Theta(\log n)$$

$A(n)$ is eventually nondecreasing and

when $n = 2^k$ $\quad A(n) = B(n) \in \Theta(B(n))$

$$A(n) \in \Theta(B(n)) = \Theta(\log n)$$

Smoothness rule

A General Divide-and-Conquer Recurrence: Master Theorem

$T(n)$ **is an eventually nondecreasing function**

$T(n) = aT\left(\frac{n}{b}\right) + f(n)$ **where** $f(n) \in \Theta(n^d), a \geq 1, b \geq 2, c > 0, d \geq 0$

$T(1) = c$ **-- General Divide-and-Conquer Recurrence**

Closed form solution: $$T(n) = n^{\log_b a}\left[T(1) + \sum_{j=1}^{\log_b n} \frac{f(b^j)}{a^j}\right]$$

$a < b^d$	$T(n) \in \Theta(n^d)$
$a = b^d$	$T(n) \in \Theta(n^d \log n)$
$a > b^d$	$T(n) \in \Theta(\boldsymbol{n}^{\log_b \boldsymbol{a}})$

Example of Using Master Theorem

$$T(n) = n^{\log_b a}\left[T(1) + \sum_{j=1}^{\log_b n} \frac{f(b^j)}{a^j}\right]$$

$T(n) = T(n/2) + 1$ ⇨ $a = 1, b = 2, f(n) = 1$

T(1)=2

⇨ $T(n) = n^{\log_2 1}\left[T(1) + \sum_{j=1}^{\log_2 n} \frac{1}{1}\right] = n^0[T(1) + \log_2 n] = 2 + \log_2 n$

Example of Using Master Theorem

$$T(n) = n^{\log_b a}\left[T(1) + \sum_{j=1}^{\log_b n} \frac{f\left(b^j\right)}{a^j}\right]$$

$T(n) = 2T(n/2) + 3n$ ⇒ $a = 2, b = 2, f(n) = 3n$

T(1)=2

⇒ $T(n) = n^{\log_2 2}\left[T(1) + \sum_{j=1}^{\log_2 n} \frac{3 * 2^j}{2^j}\right] = n^1[T(1) + 3\log_2 n]$

$= 2n + 3n\log_2 n$

Example of Using Master Theorem

$$T(n) = n^{\log_b a}\left[T(1) + \sum_{j=1}^{\log_b n} \frac{f(b^j)}{a^j}\right]$$

$T(n) = 3T(n/2) + n$ ⇒ $a = 3, b = 2, f(n) = n$

T(1)=2

$$\Rightarrow T(n) = n^{\log_2 3}\left[T(1) + \sum_{j=1}^{\log_2 n} \frac{2^j}{3^j}\right] = n^{\log_2 3}\left[T(1) + \sum_{j=1}^{\log_2 n} \left(\frac{2}{3}\right)^j\right]$$

Order of growth? $\Theta(n^{\log_2 3})$

Example of Using Master Theorem

$T(n) = T(n/2) + 1$ → $a=1, b=2, d=0,$ $a=b^d$ $T(n) \in \Theta(\log n)$

$T(n) = T(n/2)+n$ → $a=1, b=2, d=1,$ $a<b^d$ $T(n) \in \Theta(n)$

$T(n) = 2T(n/2)+3n$ → $a=2, b=2, d=1,$ $a=b^d$ $T(n) \in \Theta(n\log n)$

$T(n) = 3\,T(n/2)+n$ → $a=3, b=2, d=1,$ $a>b^d$ $T(n) \in \Theta(n^{\log_2 3})$