Friday Aug 21 '09

Iterative implementation of Recursive Algorithms

Often addresses efficiency
see HW1 will revisit later in course

Tower of Hanoi Example

Hanoi (A, B, C, n)

if n=1 then (R=1, A \rightarrow B )

else \begin{cases} 
Hanoi (A, C, B, n-1) \\
(R=n, A \rightarrow B ) \\
Hanoi (C, B, A, n-1) 
\end{cases}

\( T(n) = 2T(n-1) + 1 \)

T(1)=1

Solves to \( T(n) = 2^n - 1 \)
**Pattern**

Odd moves, in binary \( XXX1 \), \( R = 1 \) moves \textit{counterclockwise}.

Moves in binary \( XX10 \), \( R = 2 \) moves \textit{clockwise}.

Moves in binary \( X100 \), \( R = 3 \) moves \textit{counterclockwise}.

Moves in binary \( 1000 \), \( R = 4 \) moves \textit{clockwise}.

**Iterative Hanoi** \((A,B,C, n)\)

**Clockwise** := \( A \rightarrow B \rightarrow C \rightarrow A \)

**Counterclockwise** := \( A \rightarrow C \rightarrow B \rightarrow A \)

for \( i := 1 \) to \( 2^n - 1 \)

read \( i \), in binary, from right to left

\( R := \) position of first non-zero bit of \( i \)

\( \text{MOVE} (R, \text{ if } n-R = \text{even} \text{ then } \text{CLOCKWISE}) \).

\( \text{if } n-R = \text{odd} \text{ then } \text{COUNTERCLOCKWISE} \).