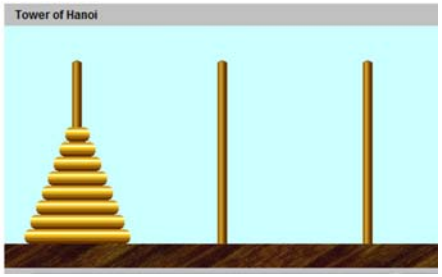
CompSci 100e
Program Design and Analysis II



February 24, 2011

Prof. Rodger

Announcements

- APT-0301 set due March 1, do 3 of 6
  - Will look at 3 of them in lab this week, but you should look at them and try them before lab.
- Next assignment is written and due March 3
  - Amortize "means" occasionally it is expensive, but usually much faster. You would apply n operations and then take the average to find the cost of one operation

Recursion

- Method calls a clone of itself
- Solves a problem by solving smaller subproblems
- "looping" by recursive calls
  - CAUTION – don't add a loop, it is implicit
- Example: see SumItUp.java
- Example: See Hanoi.java

Example: SumItUp

- Calculates and prints the sum of integers in an array
- Also prints the numbers
- For you todo: print the numbers in reverse using recursion

- Another Example: Towers of Hanoi
  - Multiple recursion
  - See Hanoi.java

Recursion (more)

- Watch out for infinite recursion
  - No way out, what happens?
  - Segmentation fault, out of memory
- Rules
  - Base case (way out) – no recursive call
  - Recursive call(s) – solve a smaller problem

Recursion vs Iteration
Which method do you use?

- Iteration
  - Easier to define
  - Faster – recursion takes some overhead
- Recursion
  - Easier to define
  - Shorter code

Types of Recursion

- Tail recursion
  - One recursive call at the end of a method
  - Easy to replace with a loop
- Reverse something
  - One recursive call “before”process
- Multiple Recursion
  - More than one recursive call

Classwork

- Recursively access directories
- Use File class
  - isDirectory() – true if file is a directory
  - Length() – size of file