

1 CS3134 #12

10/9/03

Janak J Parekh

2 Administrivia

- We have a new TA; he'll start office hours next week
- I have homeworks back
 - Still one homework with no name – come claim it!
 - Cheating problem
- Midterm: how many of you have 3 next Thursday?

3 Agenda

- Recursion

4 How to calculate...

- What's the sequence 1, 3, 6, 10, 15, 21, 28, 36...
 - *Triangle* numbers
 - How to do as loop?
 - How to do *as addition on previous result?*
 - Recursion!

5 A better example

- Simpler, you say?
- What's the sequence 1, 1, 2, 3, 5, 8, ...
 - Easy to define in terms of recursion, right?
 - How to iterate over this?
 - In other words, there are problems that are more intuitive recursively

6 Formalizing Recursion

- Recursive algorithms have the following properties
 - They call themselves
 - They call themselves to solve a smaller problem, and then work with the result
 - There's a *stopping* condition, e.g., a call which is simple enough to solve explicitly
 - Generally avoid explicit loops
- Recursion's advantages and disadvantages
 - Conceptually simpler
 - Less efficient than iteration

7 Some more examples

- FindMax
- Recursive binary search (p. 268)
- Divide-and-conquer approach
 - Take a big problem, split into smaller problems, solve separately
 - Very powerful methodology, works well with recursion

– Usually two recursive calls

8 Towers of Hanoi

- Three pegs
- Disks all on one peg
- Want to move it to third peg
- Second peg is a “work peg”
- Can’t move a disk until all smaller disks have been moved
- Basic intuition
 - Move the top disks from start to intermediate
 - Move the largest disk to destination
 - Move top disks from intermediate to destination

9 Hanoi (II)

- Three steps:
 - First, move pile from “from” to “inter”, using “to” as a work peg
 - Then, move disk from “from” to “to”
 - Then, move remainder of pile from “inter” to “to”, using “from” as a work peg
- This works because we don’t have to put things consecutively, just that larger disks must go on top of smaller disks
- Page 278 for code

10 Next time

- Today’s class is last one you need to know for midterm
- Mergesort and other sorts