1 # CS3134 #13

10/14/03

Janak J Parekh

2 # Administrivia

- Homework solutions posted
- Midterm on Thursday ☹
  - Three parts:
    - Five True/False
    - Probably four short-answer
    - One programming (may be multi-part)
  - Like a homework, but smaller – focus more on understanding concepts
  - Use my notes and solutions!

3 # Agenda

- Overloading
- Finish Recursion – mergesort
- Other sorts

4 # Overloading

- OO concept useful for recursion, but not only
- You can have multiple methods with the same name
  - As long as parameters differ
- For recursive algorithms, often will have a “bootstrap” method
- Let’s look at the FindMax example…

5 # Mergesort

- Classic recursive algorithm
- Split arrays in half, sort each half, and then merge them together
  - “Divide and conquer”
- Sort is the “recursive” call
- Let’s do it intuitively first
- Now, psuedocode…

6 # Mergesort (II)

- Key aspect of code on page 287
- The header of the method contains enough information to perform the recursive call
  - In this case, partition information
- Efficiency?
  - Partition: O(1)
  - Merge: O(n)
  - How many times each have to be done? O(log n)
  - Ergo, O(n*log n)
- Disadvantage: lots of memory required

7 # Eliminating recursion

- Recursion is often inefficient
- There are ways of eliminating it programmatically
  - Stack represents “call structure”
  - We’re not going to do this
- Often, can rethink program iteratively, if performance is needed

8 Next time

- Midterm
- After that:
  - Radix Sort
  - Quicksort