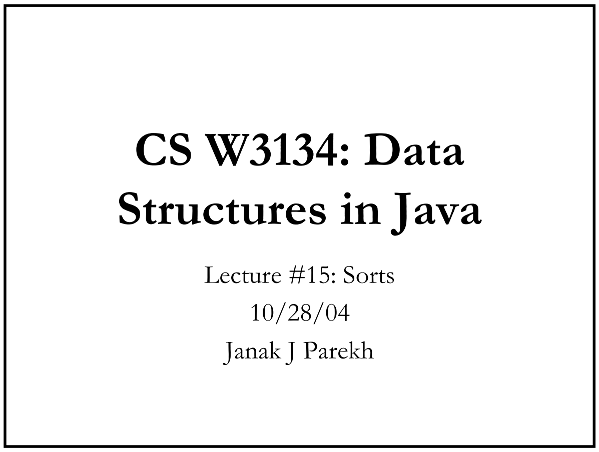
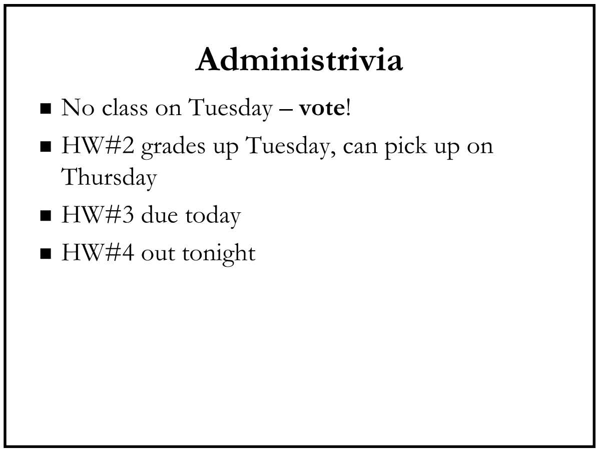


CS W3134: Data Structures in Java

Lecture #15: Sorts

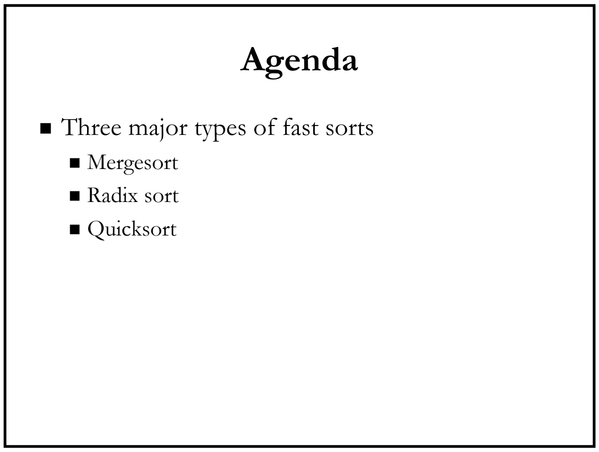
10/28/04

Janak J Parekh



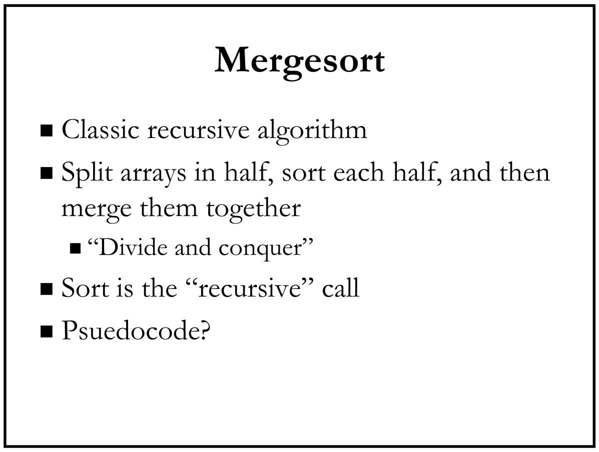
Administrivia

- No class on Tuesday – **vote**!
- HW#2 grades up Tuesday, can pick up on Thursday
- HW#3 due today
- HW#4 out tonight



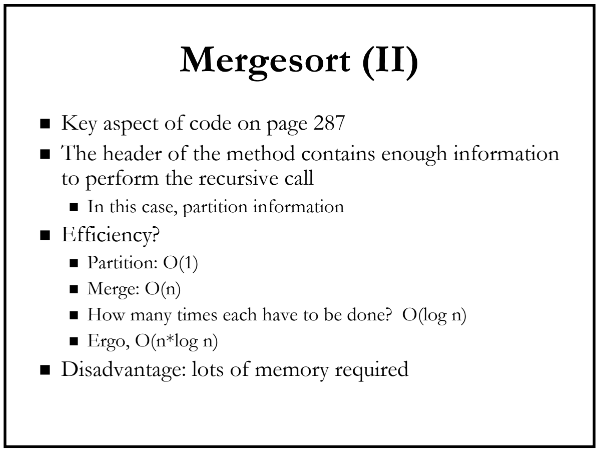
Agenda

- Three major types of fast sorts
  - Mergesort
  - Radix sort
  - Quicksort



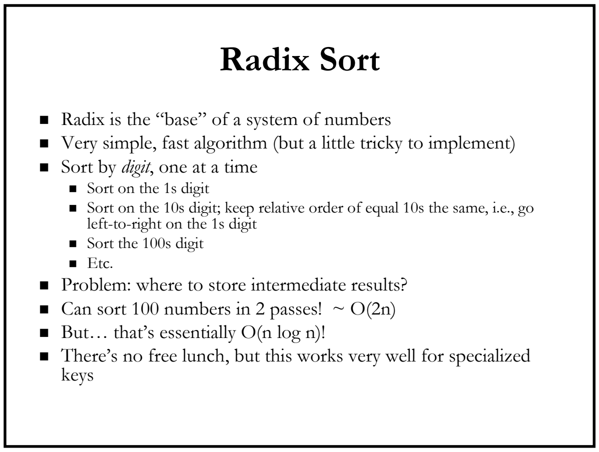
Mergesort

- Classic recursive algorithm
- Split arrays in half, sort each half, and then merge them together
  - "Divide and conquer"
- Sort is the "recursive" call
- Psuedocode?



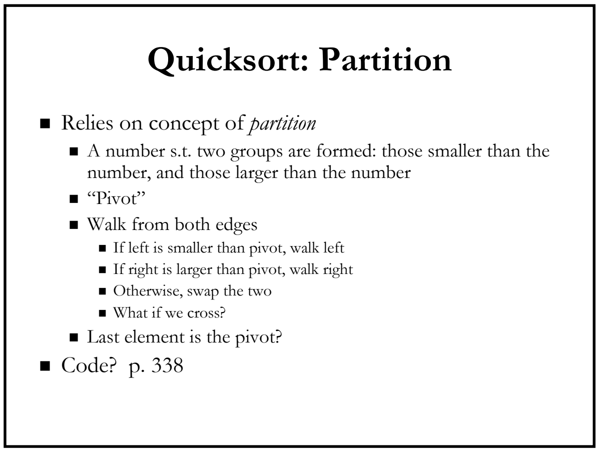
Mergesort (II)

- 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



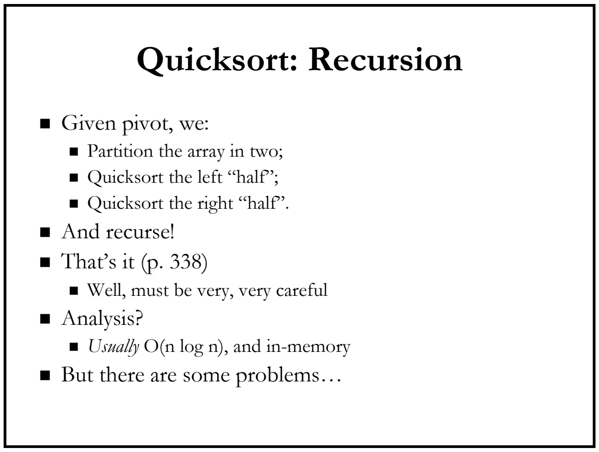
Radix Sort

- Radix is the "base" of a system of numbers
- Very simple, fast algorithm (but a little tricky to implement)
- Sort by *digit*, one at a time
  - Sort on the 1s digit
  - Sort on the 10s digit; keep relative order of equal 10s the same, i.e., go left-to-right on the 1s digit
  - Sort the 100s digit
  - Etc.
- Problem: where to store intermediate results?
- Can sort 100 numbers in 2 passes! ~ O(2n)
- But… that's essentially O(n log n)!
- There's no free lunch, but this works very well for specialized keys



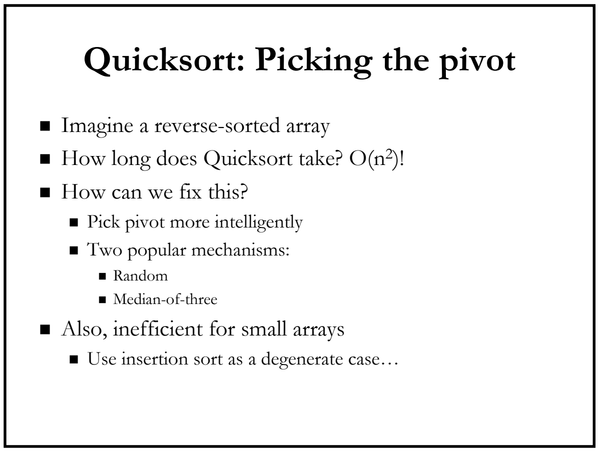
Quicksort: Partition

- Relies on concept of *partition*
  - A number s.t. two groups are formed: those smaller than the number, and those larger than the number
  - "Pivot"
  - Walk from both edges
    - If left is smaller than pivot, walk left
    - If right is larger than pivot, walk right
    - Otherwise, swap the two
    - What if we cross?
  - Last element is the pivot?
- Code? p. 338



Quicksort: Recursion

- Given pivot, we:
  - Partition the array in two;
  - Quicksort the left "half";
  - Quicksort the right "half".
- And recurse!
- That's it (p. 338)
  - Well, must be very, very careful
- Analysis?
  - *Usually* O(n log n), and in-memory
- But there are some problems…



Quicksort: Picking the pivot

- Imagine a reverse-sorted array
- How long does Quicksort take? O(n^2)!
- How can we fix this?
  - Pick pivot more intelligently
  - Two popular mechanisms:
    - Random
    - Median-of-three
- Also, inefficient for small arrays
  - Use insertion sort as a degenerate case…

Trees

- Linked Lists are generally connected to *one* other link
- What if we connect to multiple other links?
- A Tree is one generalization of a Linked List
- Key definition: no “cycles” amongst children
  - Graphs are more general
- Terminology
  - Node, Edge, Path, Root, Parent, Child, Leaf, Subtree, Level

Next time

- Start trees