CSE 373
Spring 2008
Looking Back,
Looking Forward

Today’s Outline
• Announcements
  – End of course Feedback survey posted
  – Final Exam – next Wed June 11th, 2:30-4:20
  – Ruth’s Office Hours: (if these don’t work send me email)
    • Monday June 9th 4-5pm
    • Tuesday June 10th 2-3pm
• Sorting
• Review
• Course Evaluations

Overview and Goals
(From first day handout)
Achieve an understanding of fundamental
data structures and algorithms and the
tradeoffs between different implementations
of these abstractions. Theoretical analysis,
implementation, and application. Lists,
stacks, queues, heaps, dictionaries, maps,
hashing, trees and balanced trees, sets, and
graphs. Searching and sorting algorithms.

Final Exam
• Final Exam, Wednesday June 11, 2008.
• 2:30 - 4:20pm in MGH 241 (Our regular lecture
  room)
• Exam policies
  – The exam begins promptly at 2:30pm and ends at
    4:20pm.
• The Final exam is cumulative, although a bit more
  weight will be given to topics covered since the
  second midterm.

Midterm #1
– Stacks and Queues, array and list implementations.
– Asymptotic analysis, Big-O. Worst case, upper bound,
  lower bound, analyzing loops, recurrences, amortized
  complexity.
– Trees – definitions
– Dictionary ADT
– Binary search trees – Inorder, preorder, postorder
  traversals, insert, delete, find.
– AVL trees - Single and double rotations, insert, find.
– Splay trees – insert, find, splay operations

More Computer Science Courses!!
(Fall 2008) CSE 413 Programming Languages
and their Implementation
(Winter 2009) CSE 415 Artificial Intelligence
(Winter 2009) CSE 417 Algorithms and Complexity
(Spring 2009) CSE 410 Computer Systems
  (Operating Systems & Architecture)
Midterm #2

- Binary Heaps - Findmin, Deletemin, Insert. Additional operations of increase, decrease, buildheap.
- D-heaps - Findmin, Deletemin, Insert. Additional operations of increase, decrease, buildheap.
- Leftist Heaps and Skew Heaps - Findmin, Deletemin, Insert. Additional operations of merge, increase, decrease
- Disjoint UnionFind. Up-trees. Weighted union (union by size) and path compression.
- The memory hierarchy. Temporal and spatial locality. Data structure choice and the memory hierarchy.

Since Midterm #2

- Graphs. Directed and undirected. Adjacency list and adjacency matrix representations.
- Topological sorting.
- Graph searching. Depth-first, breadth-first search.
- Sorting. Insertion sort, Selection sort, Heap sort, Merge sort, Quicksort.

Overview

• ADT – what it is, why we have them, how to compare implementations
• Comparisons – Running time, Space, Big-O, Data Locality
• Tradeoffs – Pointers, Arrays
• Algorithm Design – Iteration, Recursion, Greedy Algorithms