Course Review for Midterm Exam 1

Cpt S 223
Fall 2012
Midterm Exam 1

- **When:** Wednesday (10/10) 9:10 -10am
- **Where:** in class

- Closed book, closed notes
  - Calculators will be allowed. *No* other accessories will be allowed.
- Comprehensive
- **Material for preparation:**
  - Lecture slides & in-class notes
  - Homeworks & program assignments
  - Weiss book
Syllabus

- Comprehensive

- Topics include:
  - Math intro
  - Elementary data structures – array, list, stack, queue
  - Asymptotic notation and complexity analysis
  - Trees
  - Binary Search Trees
  - AVL Trees
More on the test…

- Generally speaking, questions in the test will tend to be modeled after classroom exercises and homework problems. But also expect to see some new types of questions.
Math Review

- Floors, ceilings, exponents and logarithms: Definitions and manipulations

- Series: Definitions, manipulations, arithmetic and geometric series closed form

- Proofs: Know definition, components, and how to use the following
  - Proof by induction
  - Proof by counterexample
  - Proof by contradiction

- Recursion
  - Know definition and rules
  - Analyze running time of recursive algorithm
  - Tail recursion removal
I won’t ask you questions in the test that are just C++ specific

C++ Review

- Know definitions and how to use the following
- Class, method, encapsulation
- Constructor, destructor, accessor, mutator
- Reference variable (&x) and call by reference
- Copy constructor, operator overloading, operator=
- Templates
- STL for different data structures
Asymptotic Analysis

- Why analyze an algorithm? TIME & MEMORY

- What do we measure and how do we measure it?
  Time: Look at the worst-case input and check for dominant terms
  Memory: Focus on the peak memory usage

- Line-by-line analysis of a program

- Best-case, worst-case and average-case analysis

- Rate of growth: Definitions and notation ($O, \Omega, \Theta, o, w$)
  - Proofs for specific examples (be familiar with the table approach)
  - Properties of asymptotic notation – e.g., symmetry, transpose symmetry, transitivity, etc.

- Maximum subsequence subproblem
Abstract Data Types

- **Lists**
  - Operations: Insert, Delete, Search
  - Implementations: vectors, singly-linked lists, double-linked lists, sentinel nodes
  - Analysis of operations for each implementation

- **Stacks** (LIFO)
  - Operations: Push, Pop, Top
  - Implementations: linked-list, vector (tradeoffs)
  - Analysis of operations for each implementation

- **Queues** (FIFO)
  - Operations: Enqueue, dequeue
  - Implementations: linked-list, vector (tradeoffs)
  - Analysis of operations for each implementation

- **Standard Template Library (STL)**
  - Use of vector, list, stack and queue template classes
  - Use of iterators
  - Know all the tradeoffs (in time & space) between all these data structures
Trees (in memory)

- **Definitions:** general m-way tree, root, leaf, child, parent, ancestor, descendant, path, height, depth

- **Binary tree:** Definition, traversals

- **Storing/representation:**
  1. All children: use array or list
  2. Store pointers to only Leftmost child and right sibling

- **Tree traversals**
  - Inorder, postorder and preorder
Search Trees

- Binary search tree (BST)
  - Definition
  - Operations: Insert, Delete, Search, FindMin, FindMax, traversals
  - Know how to perform these on a BST and show resulting BST
  - Know worst-case and average-case analysis of performance

- Balanced BST (AVL trees)
  - Definition
  - Operations: Rotations & Cases, Insert, Remove, Search, FindMin, FindMax
  - Know how to perform these on an AVL tree and show resulting AVL tree
    - Insertion – remember all four cases
    - Try out an example AVL deletion like worked out in the class (non-lazy method)
    - Know worst-case performance

- STL set and map classes (not in syllabus for Midterm Exam I)
  - Differences
  - How to use them
Good Luck !