CSE 373 -- Autumn 1999 -- Midterm Concepts

- Basic Concepts
  - ADTs - what are they, why do we use them, tensions (space vs. time, etc.)
  - math - our two favorite series, and logarithms
  - C++ templates - what they're good for

- Recursion
  - understanding of recursion
  - ability to implement recursive functions
  - understanding of benefits/overheads of recursion

- Asymptotic Analysis
  - understanding of asymptotic analysis; why it's useful
  - ability to manipulate/compare functions asymptotically
  - ability to evaluate an algorithm's asymptotic behavior (space & time)
  - best case, worst case, average case analysis
  - understanding of O(), Omega(), Theta()

- Data Structures -- for each of the following: list, stack, queue, tree, binary search tree...
  - ...understand what it is
  - ...be familiar with its major operations and their running times
  - ...know how it is implemented
  - ...have a sense for when it is useful

- Lists
  - array-based, link-based, doubly-linked, dummy head/tail nodes
  - (in)appropriateness of recursion
  - resizing array-based lists

- Stacks & Queues
  - array-based vs. link-based
  - resizing array-based implementations

- Trees
  - terminology (root, leaf, depth, ancestor, etc.)
  - fixed-degree implementation vs. first child/sibling implementation
  - pre-order, post-order traversals

- Binary Search Trees
  - numerical properties (depth vs. number of nodes vs. leaves, etc.)
  - binary search tree structure & ordering
  - pre-order, post-order, in-order traversals
  - recursive implementation of tree operations
  - lazy deletion
  - balance vs. imbalance in BSTs
  - simple rotations: why do they work?
  - AVL Trees: balance condition; maintained w/ rotations
  - Splay Trees: accessed nodes go to top; notion of amortized analysis
  - Red-Black Trees: 4 properties/intuition

- Hash Tables (I expect you to be familiar with these concepts, but won't ask questions that require intimate knowledge)
  - properties of good hash function
  - concept of string hashing
  - separate chaining
  - rehashing
  - open addressing