Information on Midterm #1

Date: Wednesday, May 5, 2010.
Type: closed book exam.

Below is the list of topics covered in this exam including the chapter and the section in the textbook where you can find the related material. **I recommend that you take as a guide lecture handouts that I provided for you since in the book there are certain things that we didn’t cover, or covered a little differently. Note that you are expected to know the material the way it is covered in the classroom.** Anything discussed in the classroom (lecture or lab) can be included in the test. At the same time, there won’t be anything in the test that was not covered in the lecture or in the lab.

Topics:

1. Algorithm Analysis (chapter 2):
   - Know general rules for computing the time function of different code segments. Be able to work out the exact formula, as well as give the big-Oh estimate for the time function of a given code segment.

2. Analysis of List operations (Lists are in Chapter 3, sections 3.2-3.5 – we just need analysis).
   - Be able to give the big-Oh estimate for the time function of every basic operation in the List ADT (insert, delete, search, check if empty) for both array and linked list implementation.

3. Stacks and Stack operations (Chapter 3, section 3.6):
   - Know what a stack is and how it can be implemented. Know basic stack operations and how they work; be able to illustrate on examples. Know both implementations of stacks – array and dynamic linked list implementations; be able to write code for basic stack operations for both implementations. Be able to write a client code to use stacks.
   - Be able to give and explain the running time estimates for different stack operations (for both, array and linked list implementations).

4. Queues and Queue operations (Chapter 3, section 3.7):
   - Know what a queue is and how it can be implemented. Know basic queue operations and how they work; be able to illustrate on examples. Know both implementations of queues – circular array and dynamic linked list implementations; be able to write code for basic queue operations for both implementations. Be able to write a client code to use queues.
   - Be able to give and explain the running time estimates for different queue operations (for both, array and linked list implementations).

5. Priority Queues and Heaps (Chapter 6, sections 6.1-6.3 including)
   - Know what a priority queue is and how it can be implemented (several ways). Be able to give running time estimates for different implementations.
   - Know what a heap is and what the heap properties are. Know different properties of complete binary trees: the index relationship between the parent and its children, the relationship between the number of nodes and the height of the tree, the relationship between the total number of nodes and the number of leaves/non-leaves in the tree.
   - Know how a heap is usually implemented (the array implementation). Know the basic heap operations and how they work; be able to illustrate on examples. Be able to write code to implement heap operations (basic operations as well as additional operations). Be able to write client code to use a binary heap.
   - Be able to give and explain the running time estimates for different heap operations.
6. Trees: Chapter 4, section 4.2.
   - Know the terminology and main concepts for trees.
   - Know the terminology and main concepts for binary trees; understand and be able to explain main characteristics of binary trees, be able to illustrate how a binary tree is implemented as a linked structure.

7. Binary Search Tree ADT: Chapter 4, section 4.3
   - Know what a BST is and what properties it has.
   - Know how a BST is implemented (a linked structure). Know the basic BST operations and how they work; be able to illustrate on examples. Be able to write code to implement BST operations (basic, as well as additional operations: find, findMinimum, findMaximum, insert, delete, isEmpty). Be able to write client code to use a BST.
   - Be able to give and explain the running time estimates for different BST operations.

**Lecture handouts related to the material covered in Midterm 1:**
- Data Structures and Abstract Data Types
- Stack ADT
- Queue ADT
- Rules for Exceptions and Exception Handling in Our Programs
- Introduction to Algorithm Analysis
- Computing the Time function and finding its tightest upper bound
- Computing the Time function and finding its tightest upper bound (MORE examples)
- Analyzing Stack, Queue, and List Operations
- Priority Queues and Heaps
- Priority Queues and Heaps (continued)
- Generic code for Comparable Objects
- Trees
- Binary Search Trees
- Implementation of basic BST operations (continued)

**Supplementary handouts containing sample codes** for the parts of following assignments:
Lab1 (main)
Lab2 (main)
Lab3 (driver and LQueue)
Lab4 (AQueue)
Lab5 (AStack)
Lab7 (TreeWork)
Project1 (MyQueue)