EECS 268 Final Exam Review Guide Spring, 2009

Administrative

- When: 7:30 - 10:00 am, Tuesday, May 12, 2009
- Where: 1136 Learned.

Warning:

Any unauthorized absence from the final exam will result in a zero grade for this exam.

Read carefully all of the following items....

- Bring your KUID. You will not be allowed to take the exam without your KUID.
- The exam will be closed book and closed notes.
- No calculators, cell phones, head phones, or electronic devices of any sort will be allowed. No such devices should be out in the open.
- Once you start the exam, you will not be excused from the room for any reason unless you turn in your exam. Once it is turned in, you cannot come back and continue working on it.

Exam Coverage

This is a comprehensive final examination and you are responsible for *all* material covered in this course. Be sure to re-examine and study the exam review guides for the first and second exams as posted earlier this semester. The remainder of this page reviews the material covered since the second exam.
Text and subject matter coverage:

- Chapters 1-12 of Carrano.
- Material covered in lectures notes 1-12 and labs.

Topic Review

Chapter 1-8

See the first two exam review guides.

Chapter 9: Algorithmic Complexity and Sorting

- Understand the concept of algorithmic efficiency and their applications.
- Understand and be able to define closed-form expression, big-O and big-Θ relations.
- Understand the basic concepts of best case, worst case, and average case complexities of algorithms.
- Understand the performance and asymptotic complexity of the sorting algorithms we studied and know the conditions under which each algorithm achieves its best, average, and worst case performances. We may ask you to supply the complexity functions for some sorting algorithms and/or also provide an input to a sorting algorithm so that the best/worst case complexity can be achieved.

- Understand and be able to describe and implement each of the sorting algorithms. We may give you a series of "snapshots" of an array while it is being sorted and ask you what algorithm is being used to sort the array.

Chapter 10: Tree and Binary Search Tree

- Understand the basic concepts and all the relevant terms and definitions of a tree.
- Understand and be able to perform preorder, inorder, postorder and level order traversals of a tree.
- Be able to use the ADT BinaryTree interface to build trees according to given specifications.
- Understand and be able to implement a binary tree using parent and children pointers.
- Understand how the *TreeNode* struct/class is used to implement various binary trees we studied.
- Understand ADT Binary Search Tree (BST) and its associated operations. Be able to describe and execute all basic BST operations. You must know the complexity of each BST operation.
- Be able to show the BST that would result from a given series of insertions and deletions.
- Understand and be able to build and/or rebuild a BST using the most efficient algorithm.

Chapter 11: Priority Queue and Heap

- Understand and be able to describe ADT PriorityQueue (PQ) and its associated operations.
- Understand the various implementation alternatives we studied (array, linked list, BST, and Heap) in implementing a PQ.
- Understand and be able to define a heap.
- Know the array-based implementation for complete binary tree and heap, including the index arithmetic required to locate the parent and children of a given node.
- Be able to describe and execute all basic heap operations. You must know the complexity of each heap operation.
- Be able to show the heap that would result from a given series of insertions and deletions.
- Understand and be able to build and/or rebuild a heap using the most efficient algorithm.

Chapter 12: Dictionary and Hash Table

- Understand and be able to describe ADT Dictionary and its associated operations.
- Be able to define hashing and describe the motivation for hashing.
- Be able to describe what a hash function is. Be able to describe what it is that characterizes a good hash function.
- Know what a collision is and be able to describe different collision resolution strategies.

- Be able to describe and execute search, insert and delete operations in a hash table. You must know the complexity of each heap operation.

- Understand the difference in performance between open and closed hashing.
- Understand hashing with chaining and be able to construct a hash table using this method.
- Understand hashing with open addressing and be able to construct a hash table using either linear or quadratic probing.
- Understand the importance of the load factor of a hash table and be able to perform rehashing.

Remark: For problems in which we give you code, be able not only to state what happens when the code executes, but also be able to locate bugs in the code.

Hints

- Be sure you understand the performance issues when implementing an ADT.
- Be sure you understand the performance issues when using various types of ADTs.
- Be sure to read and study the Key Concepts, Summary, Self-Test Exercises, and Exercises throughout the chapter. (The Programming Problems are usually less helpful when studying for exams.)
- Review the class lecture notes and lab assignments.
- Read and be sure you understand Carrano's code, especially his implementations of ADT methods.