EECS560 **Exam 2 Review Guide** **Spring, 2018**

Administrative

- **When:** 2:30pm-3:45pm, Thursday, November 8, 2018
- **Where:** 1136 Learned

Read carefully all of the following items....

- Unless otherwise stated, all material from the assigned readings, lectures and lecture notes, written and lab assignments are fair game for exams.
- 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.
- You must write legibly and show all your work clearly for credit. Partial credit will only be given to meaningful answers.
- You will be graded according to your approach to the problems, mathematical rigor, and quality of your solutions. A correct but inefficient algorithm or data structure will receive very little credit in this course.
- No make-up exam will be given for any unauthorized absence.
- Any unauthorized absence will automatically result in a zero for this exam.

Exam Coverage

Basic Computation Model and Performance Analysis

- Understand the RAM computational model and are able to use it to analyze the performance of algorithms.
- Understand the definitions and basic properties of basic asymptotic relations including big-O, big-Ω, and big-Θ.
- For a given algorithm and/or program segment, understand and are able to compute the best-case and worst-case complexity in closed-form and/or asymptotic forms using (1) detailed analysis, (2) basic operation(s), and (3) dominating steps as discussed in class.
- Understand the importance of efficient algorithms and are able to compare the performance of different algorithms and estimate their resource consumption based on their complexity functions and input size.

ADTs and Their Implementations

- In general, for a given ADT, you must be able to implement, execute, and illustrate different operations associated with the given ADT. Typical operations may include insert, findMin, findMax, find, deleteMin, deleteMax, delete, changeKey, and concate operations. You must also be able to design and analyze efficient algorithms for "other" operations using a given ADT.
- Based on the implementation of an ADT, understand and are able to design and analyze efficient algorithms for various operations related to the given ADT.
- Understand the "good" and "bad" of all ADTs and are able to compare the performance of different ADTs both analytically and experimentally.

Search Trees: Binary Search Trees (BST) & 2-3 Trees

- Understand the basic structures and properties of BST/2-3Tree class and are able to use them for various applications.
- Understand and are able to implement BST/2-3Tree using a good data structure based on their performance.
- For a given set of records, understand and are able to build a BST/2-3Tree.
- For a given search tree, understand and are able to execute and illustrate different search tree operations.
- Based on its implementation, understand and are able to design and analyze efficient algorithms for various operations related to BST/2-3Tree.

Priority Queue: K-Heaps fast deleteMin or deleteMax

- Understand the basic structures and properties of a Priority Queue (PQ) and are able to use it for various applications. complete k-ary tree and Max high-ordered tree properties
- Understand the basic structures, properties, and implementations of a k-heap.
- For a given set of records, understand and are able to build a k-heap using both the top-down and bottom-up approaches.
- For a given k-heap, understand and are able to execute and illustrate different PQ operations.
- Based on its implementation, understand and are able to design and analyze efficient algorithms for various operations related to k-heap.

DEPQ: Dual Heap and Minmax Heap

- Understand the basic structure, functionalities, and applications of the ADT DEPQ class.
- Understand and are able to illustrate the data structures used in implementing Dual Heap and MinMax Heap class.
- For a given set of records, understand and are able to build a Dual Heap and Minmax Heap using both top-down and bottom-up approaches.
- For a given DEPQ, understand and are able to execute and illustrate different DEPQ operations.
- Based on its implementation, understand and are able to design and analyze efficient algorithms for various operations related to DEPQ.

Concatenated Queues: Leftist Heap, Skew Heap, and Pairing Heap

- Understand the basic structure, functionalities, and applications of the ADT Concatenated Queue.
- Understand and are able to illustrate the data structures used in implementing Leftist heap, Skew heap, and Pairing heap.
- For a given set of records, understand and are able to build a given ADT using the most efficient algorithm.
- For a given concatenated queue, understand and are able to execute and illustrate different concatenated queue operations.
- Based on its implementation, understand and are able to design and analyze efficient algorithms for various operations related to Concatenated Queues.

10/29/2018