CISC 320 Introduction to Algorithms (Fall 2005)

Midterm Exam Study Guide

Exam Time and Date: 12:30AM to 1:45PM, Tuesday, October 18, 2005

References

- Lectures notes from start through October 11, 2005.
- Textbook: Chapters 1; 2; 3; 4.1-4.3; 6; 7; 8.1-8.3; 9.1, 9.2; 11.1-11.4;
  12.1-12.3 (excluding deletion); 13.1-13.3; 17.1-17.2; 21.1-21.3.
- Homework assignments 1 and 2.

Topics Covered

- Basic techniques for analyzing complexity and correctness of algorithms.
  - For complexity analyses: recursive tree and Masters theorem, asymptotical behavior (big Oh, Theta, and Omega), best-case, worst-case, and average-case.
  - For correctness and lower bound: induction, decision tree.

- Algorithm design paradigm: divide-and-conquer.

- Algorithms covered:
  sorting: Insertion sort, Quicksort, Mergesort, Heapsort, Counting sort, and Radix sort.
  searching: by order (Max, Min, Median, and general selection); by key (binary search, hashing)

- Data structures: Heaps, binary trees, priority queues, hash tables, red-blake trees, and disjointed sets.

Format of Exam

The exam is closed book and notes. The exam will have a heavy emphasis on understanding and applying the concepts we have discussed in class. There will be four problems.

Example questions:

- True or false, with one sentence explanation if applicable regarding the basic concepts.
  For example, in-place sorting, stable sorting, amortized time, optimality, etc.

- Describe the algorithms in concise, plain English.
  For example, quicksort.

- For the various algorithms discussed so far, give the recurrence equation (if applicable), running time for worst-case, average-case in big Oh notation. Describe the worst-case if known.

- Knowledge of lower bounds for certain problems, e.g., sorting by key comparisons.

- Evaluate recurrence equations either by using Masters theorem or by recursive tree (e.g., question 4 in homework 1).

- Draw a snapshot of some data structures after some operations. Know the basic operations for various data structures.
  For example, heaps, red-black trees, and union-find.

- Demonstrate how an algorithm works with a small input.
  For example, insert a few keys into a red-black tree.