CSE 417: Algorithms and Computational Complexity **Winter 2012**

Midterm Review Guide

The midterm will be *closed-book* and *closed-notes*. Most of the homework problems we have assigned ask you to design new algorithms or to write proofs. There may be some of this in the midterm, but because time will be limited, the questions will be more focused on your knowledge of what was presented in lecture. The questions will be shorter than homework problems, but they will require you to understand the material from class well.

List of topics

You should be able to simulate and write pseudocode for the following algorithms:

- the stable matching algorithm (page 6);
- breadth-first search (page 79 and 90); You should know how to use this for testing bipartiteness (page 94);
- depth-first search (page 83 and 93);
- testing for strong connectivity (page 98);
- topological sort (page 99);
- the greedy interval scheduling algorithm (page 118);
- the greedy interval partitioning algorithm (page 124);
- the greedy algorithm to minimize lateness (page 127);
- Kruskal's algorithm (page 143).

You should also

- be able to analyze the run time of an algorithm presented in pseudocode using asymptotic notation (similar to HW 1, Problem 3);
- understand what asymptotic notation means; know the difference between big-oh, big-omega, and big-theta, e.g., be able to explain the difference between $O(n)$, $\Omega(n)$, and $\Theta(n)$ (Chapter 2.2);
- know the properties of breadth-first and depth-first search trees; you should be able to say what kind of non-tree edges may exist for each case (Theorems 3.4 and 3.7 in the book);

- understand the *Cut Property* (Theorem 4.17) and the *Cycle Property* (Theorem 4.20);
- know the Master's theorem (Recurrences handout); this should be something you have memorized;
- construct counterexamples to show when greedy algorithms do not work (e.g. HW 3, Problem 2).

Practice Problems

In addition to going through the material presented in lecture, one of the best ways to prepare for the midterm is to go through the homeworks and make sure you understand the solutions. Here we're giving you some additional problems that we think will help you study for the midterm. We'll go through some solutions during the review session.

1. Solve the following recurrences to get the best asymptotic bounds you can on $T(n)$ in each case using $\Theta()$ notation. Assume all numbers are rounded to the nearest integer.
   (a) $T(n) = 2T\left(\frac{n}{8}\right) + n^{0.4}$, and $T(1) = 1$. Assume n is a power of 8.
   (b) $T(n) = 7T\left(\frac{n}{7}\right) + \frac{n}{3}$, and $T(1) = 1$. Assume n is a power of 7.
   (c) $T(n) = 5T\left(\frac{n}{2}\right) + n^2$, and $T(1) = 1$. Assume n is a power of 2.
2. State whether each claim is true or false, and give a short justification for your answer.
   (a) Any function that is $O(n)$ is also $O(n^2)$.
   (b) Any function that is $O(n^2)$ is also $O(n)$.
   (c) If f is the best-case complexity of an algorithm and g is its worst-case complexity then $f(n)$ is always $O(g(n))$.
   (d) There is a function that is both $O(n \log n)$ and $\Omega(n)$
   (e) There is a function that is both $O(n)$ and $\Omega(n \log n)$
   (f) It is possible for an algorithm to run in time $O(n)$, and for its measured run times on inputs of length 0, 1, 2, 3, 4, 5, 6, 7 to be 1, 2, 4, 8, 16, 32, 64 and 128 seconds, respectively.
3. Chapter 3, Exercise 1 (page 107).