CSE 521: Design and Analysis of Algorithms — Autumn 2006

Problem Set #1 — Instructor: Anna Karlin

Due on **October 12, 2006** in class.

Reminder: If you haven't done so already, subscribe to CSE 521 email group ASAP by following the link from the course webpage `http://www.cs.washington.edu/521`.

Instructions: You are allowed to collaborate with fellow students taking the class in solving problem sets. You may also collaborate with one other classmate on writing up your solutions. If you do collaborate in any way, you must acknowledge for each problem the people you worked with on that problem.

The problems have been carefully chosen for their pedagogical value and hence might be similar or identical to those given out in past offerings of this course at UW, or similar courses at other schools. Using any pre-existing solutions from these sources, from the Web or other algorithms textbooks constitutes a violation of the academic integrity expected of you and is strictly prohibited.

Most of the problems require only one or two key ideas for their solution – spelling out these ideas should give you most of the credit for the problem even if you err in some finer details. So, make sure you clearly write down the main idea(s) behind your solution even if you could not figure out a complete solution.

A final piece of advice: Begin work on the problem set early and don't wait till the deadline is only a few days away.

Readings: Kleinberg and Tardos: Chapter 1; Chapter 4 (till 4.7); Chapter 5.

Each problem is worth 10 points unless noted otherwise. All problem numbers refer to the Kleinberg-Tardos textbook.

1. Chapter 1, Problem 4
2. Chapter 1, Problem 6
3. Chapter 1, Problem 8 (Truthfulness in Gale-Shapley algorithm)
   Suggestion: Try to work out some small examples to get started.
4. Chapter 4, Problem 2
5. Chapter 4, Problem 13
6. Consider the problem of making change for n cents using the least number of coins.
   (a) Describe a greedy algorithm to make change consisting of quarters, dimes, nickels, and pennies. Prove that your algorithm yields an optimal solution.
   (b) Suppose that the available coins are in the denominations $b^0, b^1, \ldots, b^k$ for some integers $b > 1$ and $k \geq 1$. Show that the greedy algorithm always yields an optimal solution.
   (c) Give a set of coin denominations for which the greedy algorithm does not yield an optimal solution.