CSE 521: Design and Analysis of Algorithms — Autumn 2006
Problem Set #5 — Instructor: Anna Karlin
Due on **December 11, 2006**

Turn-in: You can either email your homework to Roee or leave it in his mailbox on the first floor of the Allen Center (in the room with all the CSE grad student mailboxes).

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 **important** 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 8 and chapter on linear programming here:
`www.cse.ucsd.edu/users/dasgupta/mcgrawhill/chap7.pdf`

Each problem is worth 10 points unless noted otherwise. All problem numbers refer to the Kleinberg-Tardos textbook. On the NP-completeness proofs, you can use any NP-complete problem discussed in the chapter for your reductions.

1. Chapter 8, Problem 9
2. Drawing graphs nicely is a problem that arises constantly in applications. Consider the problem of drawing a tree. Some characteristics that would be desirable in the drawing are:
   - All nodes on the same level in the tree should line up horizontally.
   - The vertical distance between a node and its children in the tree should not be less than some minimum value m.
   - All nodes should lie within a certain window on the screen.
   - The parent of a set of nodes should be centered over those nodes in the horizontal direction.
   - The height and width of the tree drawing should be small.

   How would you formulate the problem of placing the tree nodes in the drawing using linear programming? (The problem statement has purposefully been left somewhat vague. It is up to you to formalize both the problem and the solution.)

3. A multicommodity flow network supports the flow of p different commodities between a set of p source vertices $S = \{s_1, \ldots, s_p\}$ and p sink vertices $T = \{t_1, \ldots, t_p\}$. For any edge (u, v) the net flow of the ith commodity from u to v is denoted $f_i(u, v)$. For the ith commodity, the only source is s_i and the only sink is t_i. There is flow conservation independently for each commodity: the net flow of each commodity out of each vertex is zero unless the vertex is the source or sink for the commodity. The sum of the net flows of all commodities on an edge (u, v) must not exceed the capacity of the edge $c(u, v)$, and in this way the commodity flows interact. The value of the flow of each commodity is the net flow out of the source for that commodity. The *total flow value* is the sum of the values for all p commodity flows.

   - Give a linear programming formulation for maximizing the total flow value in a given multicommodity flow network.
   - Give the dual program.

4. Use the simplex algorithm to solve the following linear program:

   maximize $3x_1 + 2x_2 + 4x_3$ subject to the constraints:

   $x_1 + x_2 + 2x_3 \leq 4$

   $2x_1 + 3x_3 \leq 5$

   $2x_1 + x_2 + 3x_3 \leq 7$

   $x_1, x_2, x_3 \geq 0.$

   Show the steps of the algorithm.

5. Use duality to prove the following theorem:

   Let M be an m by n matrix. Let $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ represent probability vectors (i.e. $\sum_i p_i = \sum_j q_j = 1$ and all p_i and q_j nonnegative).

   Then
   $$\max_{\mathbf{p}} \min_{j} \sum_{1 \leq i \leq m} p_i M_{ij} = \min_{\mathbf{q}} \max_{i} \sum_{1 \leq j \leq n} M_{ij} q_j.$$

   Remark: This is the von Neumann minimax theorem for 2-person zero sum games.