Algorithms — Winter 2007

Problem Set #6 — Instructor: Anna Karlin

Due on **Feb 26, 2007** by 7pm. (**note the date change**)

Instructions: You are welcome to brainstorm on ideas for solving these problems with fellow students taking the class. You may also collaborate with one other classmate on writing up your solutions. If you do collaborate in any way, please acknowledge for each problem the people you worked with on that problem. Please do not post solution ideas to discussion boards – discussion boards can be used to clarify the problems and perhaps to discuss specific examples. Learning will not be effective if people can find solutions (even partial solutions) to the problems on the dicussion board, on the Web or in other algorithms textbooks.

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.

Be sure to carefully read the grading guidelines page linked off the course web page.

Readings: Kleinberg and Tardos: Chapters 8 and
`http://www.cs.berkeley.edu/ vazirani/algorithms/chap7.pdf`

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

1. 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.)

2. 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 multi-commodity flow network.

3. 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
$$\max_{\mathbf{p}} \min_{j} \sum_{1 \le i \le m} p_i M_{ij} = \min_{\mathbf{q}} \max_{i} \sum_{1 \le j \le n} M_{ij} q_j.$$

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

4. Chapter 8, Problem 3