CS G713: Advanced Algorithms	**Problem Set 4**
Assigned: November 7, 2008	Due: November 21, 2008 before class

Guidelines If you write "I do not know how to solve this problem" then you get 1/4 of the score for the problem. If you write nonsense then you get 0.

As we are going to learn in this class, *time* and *space* are very valuable resources. Strive to give effective, compact solutions. Your solutions should touch on all the main points, but long calculations or discussions are not required nor sought.

Do not worry if you sometimes "do not get it." The problems are meant to stimulate you, not to overwork you. Unless specified otherwise, you can collaborate, but you must acknowledge all your collaborators in your solutions. To hand in your solutions: Give it to me, slide it under my door West Village H (246), or email it to csg713-instructor@ccs.neu.edu.

Problem 1. von Neumann's minmax theorem Let A be an $n \times n$ matrix. Prove:

$$\min_x \max_y yAx = \max_y \min_x yAx,$$

where the maximums and minimums are taken over probability distributions x, y, i.e., $x, y \in R^n$ such that $x_j, y_j \geq 0$, for every j, and $\sum_j x_j = \sum_j y_j = 1$). (We can interpret x and y as probability distributions for two players (i) and (ii) over their set of strategies $\{1, \ldots, n\}$ – x, y are called mixed strategies – and A as a payoff matrix. yAx is then the expected payoff when (i) plays x, and (ii) plays y. In this interpretation the theorem asserts that if (i) has a mixed strategy x that achieves an expected payoff of at most t no matter what (ii) plays, than (ii) has a mixed strategy y that achieves payoff at least t no matter what (i) plays.)

Hint: Rewrite the desired equation so that each side of the equation involves only one min or max (not an alternation as is currently).

Problem 2. Corollary to Farkas' lemma Assume Farkas' lemma:
$\exists x : Ax = b, x \geq 0$ if and only if there is no $y : A^T y \geq 0, b \cdot y < 0$.

Prove the following corollary:
$\exists x : Ax \leq b$ if and only if there is no $y : A^T y = 0, y \geq 0, b \cdot y < 0$.

Note: For the next two problems we are counting *bit* operations. You can however assume the following: (1) arithmetic between t-bit integers takes time poly(t), and (2) the multiplication of two polynomials of degree n with coefficients in $\{0, 1\}$ can be performed in $O(n \cdot \text{poly} \log n)$ bit operations. ((2) follows from Fast Fourier Transform techniques.)

Problem 3. Fast hash functions Recall the following family of hash functions $h_a : \{0,1\}^n \to \{0,1\}^n$ from PS2: For a random $a \in \{0,1\}^{2n-1}$,

$$h_a(s) := (\sum_{i=1}^{n} a_{n-1+i} \cdot s_i, \sum_{i=1}^{n} a_{n-2+i} \cdot s_i, \ldots, \sum_{i=1}^{n} a_{1+i} \cdot s_i, \sum_{i=1}^{n} a_{0+i} \cdot s_i),$$

where the sum in each entry is modulo 2, and so the output of h is in $\{0,1\}^n$.

Show that given a, s, the n-bit output $h_a(s)$ can be computed in $O(n \cdot \operatorname{poly}\log n)$ bit operations.

Problem 4. Fast integer multiplication Show how to compute the product $a \cdot b$ of two n-bit integers a, b in $n \cdot \operatorname{poly}\log n$ bit operations.