CS G713: Advanced Algorithms	**Problem Set 0**
Assigned: September 12, 2008	Due: September 16, 2008 before class

This problem set is optional, but you are encouraged to solve it.

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. (Also keep in mind that there is always a non-zero chance that I mess up something in a problem ;-) You do *not* have to solve all the problems to get an 'A' in this class. As a rule of thumb, skipping a problem per problem set is enough to an 'A' in this class.

In general you can collaborate, but you must acknowledge all your collaborators in your solutions. On selected problem sets (e.g. the final) you cannot collaborate and you should think about the problems yourselves.

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. Probability recap. (1) Prove the Cauchy-Schwarz inequality: For every random variable X,

$$E[X^2] \geq E[X]^2.$$

(2) Conclude that for any vector $v = (v_1, \ldots, v_n) \in \mathbb{R}^n$ we have

$$\sum_i v_i \leq \sqrt{n} \cdot \sqrt{\sum_i v_i^2}.$$

(3) The *statistical distance* between two distributions X, Y is the maximum over all events T, of $|\Pr[X \in T] - \Pr[Y \in T]|$. Prove this equals $\frac{1}{2} \cdot \sum_a |\Pr[X = a] - \Pr[Y = a]|$.

Problem 2. Algorithm recap (1) Let A be a randomized algorithm for a Boolean problem. Suppose that on YES instances the algorithm is always correct, but on NO instances the algorithm errs with probability at most 1/2. Derive a new algorithm A' that is always correct on YES instances, and on NO instances errs with probability at most $1/2^{1000}$. How slower is the new algorithm?

(2) Let A be a deterministic algorithm that runs in space (or memory) $t \geq \log n$. Argue that the algorithm runs in time $2^{O(t)}$.