6.896 Sublinear Time Algorithms	October 28, 2004
Homework 2	
Lecturer: Ronitt Rubinfeld	*Due Date: November 16, 2004*

Homework guidelines: Same policy as last time. Remember to turn in your solution to *each* problem on a separate sheet of paper, with your name on each one.

Here are some problems to think about, but don't turn them in.

- In Alon's proof of the lower bound for testing the property of triangle-freeness, show that the number of edge-disjoint triangles is in the s-blow up of the graph is at least $\Omega(m|X|s^2)$ (see lecture 9 notes for definitions of all variables).
- You are given an approximation scheme $\mathcal{A}$ for f such that on input x, ϵ, $Pr[f(x) - \epsilon \leq \mathcal{A}(x) \leq f(x) + \epsilon] \geq 3/4$, and $\mathcal{A}$ runs in time polynomial in $1/\epsilon, |x|$. Construct an approximation scheme $\mathcal{B}$ for f such that $Pr[f(x) - \epsilon \leq \mathcal{B}(x) \leq f(x) + \epsilon] \geq 1 - \delta$, and $\mathcal{B}$ runs in time polynomial in $\frac{1}{\epsilon}, |x|, \log \frac{1}{\delta}$.

The following problems are for turning in (and maybe also for thinking about).

1. (Canonical forms for graph property testers for the adjacency matrix model). Define a graph property to be a property that is preserved under graph isomorphism – i.e., if G has the property and G' is isomorphic to G, then G' must also have the property. Prove the theorem cited in class due to Goldreich and Trevisan, namely:

   Theorem 1 *Let P be any graph property in the adjacency matrix model. Let T be a property tester for P that on input G, ϵ, where G is a graph on n nodes, has query complexity $q(n, \epsilon)$. Then there is a (nonadaptive) tester T' that selects a random subset of $2q(n, \epsilon)$ nodes in G, queries the edges between every pair of the subset and decides whether to pass or fail based on the query answers and its own internal coin tosses. Furthermore, if T has 1-sided error, then so does T'.*

   It is interesting to make sure you can first solve the following, even though the solution will look nothing like your eventual solution (and don't bother to turn it in). Show that in the adjacency matrix model, any algorithm making q queries can be made into a *nonadaptive* (i.e., where the queries do not depend on the results of any previous queries) tester that uses only 2^q queries.

2. (This is mainly an exercise in using indicator functions and Chernoff bounds). The following questions concern the max cut algorithm, due to Goldreich, Goldwasser and Ron, presented in class.

   (a) In the analysis of the "main claim", we defined U^i to be "good" if for at least $(1 - \epsilon/8)|V^i|$ of the nodes $v \in V^i$, the $(*')$ conditions holds, namely that $\forall j \in \{1, 2\}, |\Gamma(v, U^i_j)/t - \Gamma(v, W_j)/|W|| \leq \epsilon/32$ (where $W = W_1 \bigcup W_2$). Show that the probability that U^i is good is at least $1 - 1/6\ell$. *Note that the (*') condition is slightly different than the condition given in class. Also, if you find it helpful, you may modify the definition of ℓ as long the dependence on ϵ is still $\theta(1/\epsilon)$.*

(b) Give an approximation algorithm for max-cut on dense graphs which uses a number of queries that is independent of the number of nodes in the graph. You may use any theorem or algorithm that we proved in class as a starting point. There is one additional algorithmic idea that makes the analysis particularly simple given what we have already proved.

3. (Property testing of the clusterability of a set of points.) Given a set X of points in any metric space. Assume that one can compute the distance between any pair of points in one step. Say that X is (k, b)-diameter clusterable if X can be partitioned into k subsets (clusters) such that the maximum distance between any pair of points in a cluster is b. Say that X is ϵ-far from (k, b)-diameter clusterable if at least $\epsilon|X|$ points must be deleted from X in order to make it (k, b)-diameter clusterable.

   Show how to distinguish the case when X is (k, b)-diameter clusterable from the case when X is ϵ-far from $(k, 2b)$-diameter clusterable. Your algorithm should run in time polynomial in $k, 1/\epsilon$. It is possible to get an algorithm which runs in time $O((k^2 \log k)/\epsilon)$.

   Note that the [ADPR] result presented in class works only for a specific metric space, whereas this algorithm should work in general.