CSC 2420F 2010, Assignment 3
Due: December 17, AM

1. We are given a degree bound d and query access to a partial table for a function $f : Q \rightarrow Q$; namely given $\{(x_1, f(x_1), \ldots, (x_n, f(x_n)\}$ we can access any $(x_i, f(x_i))$ in one query. We consider the following questions for $d << n$:

   (a) We want to test if the partial table f is equal to a degree d polynmial p or if it is "far-away" from any degree d polynomial where by far-away we mean that $f(x_i) = p(x_i)$ for at most $(1 - 2/d)n$ of the points given in the table. Provide a randomized 1-sided error algorithm that will make $O(d)$ queries, always returning p if it exists and with probability $\geq \delta$ will determine that f is far-away from any degree d polynomial. If neither condition is true, the algorithm can give any answer.

   Analyze the probability δ that can be achieved in terms of the number of queries used.

   (b) (Small Bonus) We now want to test if the partial table f is "close-to" a degree d polynomiial p or "far-away" where far-away is as before and "close-to" means that $f(x_i) = p(x_i)$ for at least $(1 - 1/d)n$ of the points given in the table. Provide a randomized 2-sided error algorithm that will make $O(d)$ queries, returning p if it exists with probability $\geq \delta$ or determining with probability $\geq \delta$ that f is far-away from any degree d polynomial. If neither condition is true, the algorithm can give any answer.

2. In the bounded degree graph model, provide a sublinear (in n) time algorithm that with high probability will distinguish graphs having only small connected components (for definiteness say at most size $n^{1/8}$) from graphs having a large component (for definiteness say size at least $n^{1/2}$). To elaborate, n is the number of nodes of the graph being tested, and the goal is to always say YES if the graph only has small components, and to say NO with at least constant probability if the graph has at least one large component.

3. We saw that a random walk on a line can be used to derive an expected time $O(n^2)$ randomized (1-sided) algorithm for 2-SAT. Consider now

the extension of this algorithm for k-SAT. Using elementary probability show that for every k, there exists $c_k < 2$ such that a randomized algorithm can solve $k - SAT$ in expected time $(c_k)^n$.

4. (a) Consider the randomized LP rounding algorithm for weighted Max-Sat that achieves a β approximation where $\beta \geq 1 - 1/e$. Describe how to de-randomize this algorithm.

   (b) Indicate whether or not there is an obvious way to transform this algorithm into a greedy (i.e. priority) algorithm similar to how we can de-randomize the naive randomized algorithm for exact Max-k-Sat so that it becomes a priority (even online) algorithm. That is, consider the input items to be the propositional variables where each variable is represented by a full description of the clauses in which the variable appears.

   (c) (Bonus question) For the exact Max-2-Sat problem can you prove any positive or negative results? That is, does there exist a greedy (i.e. priority) or local search algorithm that improves upon the $\frac{3}{4}$ i approximation achieved by the de-randomized naive randomized algorithm or the non-oblivious local search algorithm? Can one prove any inapproximation result for such algorithms?
   Hint: This is mainly a problem that you might want to think over during the holidays with a good glass of wine. If you are successful, I would think this would be the basis for a good publication. But at least you will enjoy the wine. There is an implied (weak) inapproximation result in a paper by Alekhnovich et al for a model that generalizes priority algorithms but I am looking for better inapproximation than that result implies.

5. Possibly more to follow