CSC 2420 Spring 2016, Assignment 3
Due date: April 18, noon

NOTE: If you are taking the course for credit, then you may either work by yourself or with at most one other student taking the course for credit. You must specify with whom you are collaborating and the extent of collaboration. It is certainly preferable for you to solve the questions without consulting a published source. However, if you are using a published source then you must specify the source and you should at least try to improve upon the presentation of the result.

1. Recall that Yanakakis showed that Johnson's Max-Sat algorithm was the derandomization of the naive randomizaed algorithm. The naive algorithm and Johnson's algorithm can both be viewed as online algorithms where the naive algorithm can be implemented in what we called input model 0 whereas Johnson's algorithm can be implememted in what we called input model 1. See Lecture 7. Yannakakis also showed that Johnson's algorithm is no better than a $\frac{2}{3}$ approximation. Yanakakis' (non exact) 2-Sat nemesis input consists of three clauses; namely, $x \vee y$, $x \vee \bar{y}$, and then either clause y or clause $\bar{y}$. Variable x arrives first in the online order. The following inapproximation results concerning max-of-2 online algorithms are due to Nicolas Pena and in fact hold for any max-of-k online algorithms for any constant k. By an α inapproximation, we mean that we cannot do better than obtaining an α approximation ratio (or sometimes to be more precise we mean than we cannot achieve $\alpha + \epsilon$ for any $\epsilon > 0$). By a max-of-k online algorithm for max-sat, we mean a collection of k deterministic online algorithms for max sat (eg Johnson's algorithm is an online algorithm) each making possibly different decisions (true or false) for each propositional variable (and each facng the same online sequence of variables). Then take the best solution.

   (a) Complete the Yannakakis argument showing the $\frac{2}{3}$ inapproximation even for input model 3.

   (b) Prove a $\frac{3}{4}$ inapproximation with respect to input model 2 for any $max-of-2$ online algorithm for exact Max-2-Sat.

Hint: The nemesis input will consist of three propositional variables, x_1, x_2 and y with variable y occuring last again in the input order.

(c) (Bonus) Prove a $\frac{5}{8}$ inapproximation with respect to input model 0 for any $max - of - 2$ online algorithm for non exact 2-Sat. Note: As of now the proof for this inapproximation needs repeated clauses. The $\frac{5}{8}$ inapproximation is clearly than the $\frac{2}{3}$ approximation achieved by Johnson's algorithm with respect to input model 1. Johnson's algorithm (and all the algorithms presented for Max-Sat problems work whether or not there are repeated clauses. But it would be more satisfying if an inapproximation result did not use repeated clauses.

2. Recall the Yannakakis randomized rounding algorithm for Max-Sat that achieves a $1 - \frac{1}{e}$ expected approximation. This can also be coupled with the naive randomized algorithm to achieve a $\frac{3}{4}$ expected approximnation ratio.

   - Desribe how you would derandomize the randomized rounding algorithm to achieve a deterministic $1 - \frac{1}{e}$ approximation for Max-Sat.
   - In the case of Max-k-Sat, we indicated how the derandomization could be implemented by an online algorithm in the simplest input model for Max-Sat where each propositional variable x is represented by the name, weight and length of each clause in which x appears as a literal and the same for $\bar{x}$. Do you think the Yannakakis algoerithm can be implemented by a deterministic online algorithm in such an input model? Explain your answer.

3. We are given a degree bound $d << n$ and query access to a partial table for a function $f : Q \to Q$; 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. Consider the following:

   (a) We want to test if the partial table f is produced by 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) 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.

4. Give an informal but convincing argument to complete the probabilistic analysis for the sublinear time algorithm for searching in an anchored sorted linked list. See Lecture 10.

5. Consider the proof of the approximation ratio provided by the 1-exchange local search algorithm for maximizing a monotone submodular function subject to a matroid constraint. See slides 28,29,30 in Lecture 11.

   (a) Prove the first fact about submodular functions that appears on slide 28.

   (b) Fill in the details for the concluding inquality on slide 30 (using whatever other facts, inequalities) are given in the proof.

   (c) Show precisely where monotonicity is being used in the proof of the approximation ratio. Conclude that if the submodular function f was α monotone (for $\alpha \geq 1$) then the 1-exchange algorithm would provide a $\frac{1}{2\alpha}$ approximation. Here I define α monotone by the condition:

$$f(S) \leq \alpha f(T) \quad \forall S \subseteq T$$

.