CSC373F **Problem Set 3 Selected Solutions)** **Fall, 2011**

Due: Wed, November 30, beginning of lecture

NOTE: Each problem set only counts 5% of your mark, but it is important to do your own work (but see below). Similar questions will appear on the next term test. You may consult with others concerning the general approach for solving problems on assignments, but you must write up all solutions entirely on your own. Anything else is *plagiarism*, and is subject to the University's Code of Behavior. You will receive 1/5 points for any (non bonus) question/subquestion for which you say "I do not know how to answer this question". You will receive .5/5 points if you just leave the question blank.

1. (10 points)

   Let L be a NP complete set and let one NP representation of L be $L = \{w|\exists y[R(w,y)$ is true and $|y| \leq q(|w|\}$. Here q is a polynomial and R s a polynomial time predicate. The associated search problem, $L(w,R,q)$, is "Given w, find a certificate y if it exists (i.e. if $w \in L$), else say that y does not exist". Show that the search problem $L(w,R,q)$ can be polynomial time reduced to the decision problem for L. That is, $L(w,R,q) \leq_T^{poly} L$.

   SOLUTION: Our goal is to define another NP set L' : $L(w,R,q) \leq_T^{poly} L'$. (Since L is NP-complete, $L' \leq_m^{poly} L$ and hence $L(w,R,q) \leq_T^{poly} L$ since $\leq_m^{poly}$ implies $\leq_T^{poly}$ and $\leq_T^{poly}$ is transitive. (We also note in passing that L' will be NP-complete since L is NP-complete.) We will assume that $L \subseteq \{0,1\}^*$.

   Let $L' = \{(w,z)|\exists z' : R(w,zz')$ and $|zz'| \leq q(|w|)\}$. That is, $(w,z) \in L'$ if z is a prefix of a certificate for $w \in L$. By definition, L' is an NP set. It remains to show that $L(w,R,q) \leq_T^{poly} L'$. The reduction is via the following program:

   $z := \epsilon$ % ϵ is the null string of length 0
   If $(w,z) \notin L'$ then report $y \notin L$ and halt
   Else % We we know there is a certificate and z will become the certificate
   While $R(w,z)$ not true
   If $R(w,z0) \in L'$, then $z := z0$; else $z := z1$
   End IF
   End While

   If we wanted a proof that this does yield the required reduction, we can prove by induction on $|z|$, that z is a prefix of a certificate y. The reduction (i.e. the algorithm) above creates the certificate one bit at a time and as defined creates the lexicographically first certificate.

   NOTE: In the original formulation, it is sufficient that L be in NP. But in the new formulation, it is important that L also be complete. In particular, we could

formulate the set $COMPOSITE = \{w|\exists y : y \text{ is a proper divisor of } w\}$ where w and y say are decimal representations of positive integers. Whereas we know that $COMPOSITE$ is in P, the search problem essentially solves the integer factoring problem which we strongly believe cannot be computed in polynomial time.

2. Consider the following *3 from 4 frequency set cover problem*: We are given a collection of sets $\mathcal{S} = \{S_1, \ldots, S_m\}$ for $S_i \subseteq U$ with the property that every $u \in U$ occurs in exactly four different sets S_i in $\mathcal{S}$. There is also a cost function $c : \mathcal{S} \to \mathbb{R}^{\geq \nvDash}$ and we let c_i denote the cost of set S_i. A feasible solution is a sub-collection $\mathcal{S}' \subseteq \mathcal{S}$ such that every $u \in U$ occurs in at least three different sets S_i in $\mathcal{S}'$. The goal is to find a feasible solution $\mathcal{S}'$ so as to minimize the cost $c(\mathcal{S}') = \sum_{i:S_i \in \mathcal{S}'} c_i$.

   (a) Formulate the *3 from 4 frequency set cover problem* as a $\{0,1\}$ IP

   (b) Show how to use LP relaxation + rounding to obtain a 2-approximation algorithm. Explain why your rounded solution is a feasible solution to the IP and why it provides a 2-approximation.

   SOLUTION: As defined in the Friday, Dec 2 lecture, the IP is minimize $\sum c_i x_i$ subject to
   $\sum_{i:u \in S_i} x_i \geq 3$ %one such constraint for each $u \in U$
   $x_i \in \{0,1\}$

   The LP replaces $x_i \in \{0,1\}$ by $x_i \in [0,1]$

   Let $\{x_i^*\}$ be an optimal LP solution. Define $x_i' = 1$ if $x_i^* \geq 1/2$; 0 otherwise

   It remains to show that the $\{x_i'\}$ are an integral (IP) solution and that it is a 2-approximation. Since every $x_i' \leq 2$ is is clear that $\sum c_i x_i' \leq 2 \sum c_i x_i^* = LP{-}OPT \leq OPT$.

   It remains to show that the $\{x_i'\}$ are an IP solution. Let $u \in U$. Say for ease of presentation that $u \in S_1, S_1, S_3, S_4$.

   Suppose that $\sum_{i:u \in S_i} x_i' < 3$ and hence at most 2. Say for ease of presentation that $x_1' = x_2' = 0$ and hence that $x_1^* < 1/2$ and $x_2^* < 1/2$. Then $x_1^* + x_2^* + x_3^* + x_4^* < 3$ and hence $\{x_i^*\}$ was not an LP solution.

3. Suppose we are given an algorithm that multiplies two degree n univariate polynomials $a(x)$ and $b(x)$ and returns what is supposed to be the degree $2n$ polynomial $c(x) = a(x) * b(x)$. Describe an $O(n)$ time randomized testing algorithm and explain the error bound in terms of the time bound. More precisely, your testing algorithm will always say "good" if indeed $c(x) = a(x) * b(x)$ and will say "bad" with some constant probability $\delta > 0$ if $c(x) \neq a(x) * b(x)$.

   SOLUTION: There was some ambiguity in the way this question was phrased so lets say we choose a sample set S of size $4n$, and lets say that to evaluate a degree r polynomial at a point (in S) costs r (scalar) operations. Since $a(x) * b(x)$ has degree

$2n$, we can assume that the polynomial $d(x) = c(x) - a(x) * b(x)$ has degree at most $2n$. The randomized testing algorithm is to compute $d(x)$ at a random point $s \in S$. If $d(s) \neq 0$ then output "bad". Else output "good". The number of operations to compute $d(s)$ is at most $4n + 2$ by our assumption that evalution of a degree r poly takes r operations.
Claim: If $c(x) = a(x) * b(x)$ then the testing algorithm does not error (always output "good") and if $c(x) \neq a(x) * b(x)$ then the probability that the testing algorithm will error (i.e. will output "good") is at most $1/2$. This is because a non zero polynomial of dgeree $2n$ has at most $2n$ distinct zeros and we are sampling from $4n$ points. If we want error δ then we need to run this testing algorithm t times independently so that $(1/2)^t \leq \delta$; i.e. $t \geq \log(1/\delta)$ at a cost of $t * (4n + 2)$ operations.