CSE 431 – Theory of Computation. Spring, 2014. Instructor: James R. Lee

ASSIGNMENT 5. Due Thursday, May 15th, in class (or via email to cse431-staff@cs before class starts)

1. Consider the language
   TRIPLE-SAT $\{\langle\phi\rangle : \phi$ is a Boolean formula that has at least three satisfying assignments $\}$
   Show that TRIPLE-SAT is NP-complete using a reduction from SAT (which we know is NP-complete).

2. A **vertex cover** in an undirected graph $G = (V, E)$ is a subset of nodes $C \subseteq V$ such that every edge has at least one endpoint in C. A **dominating set** of G is a set of nodes $D \subseteq V$ such that every node of G is in D or has a neighbor in D. Consider the two problems:
   VERTEX-COVER $= \{\langle G, k\rangle : G$ has a vertex cover of size at most $k\}$
   DOMINATING-SET = $\{\langle G, k\rangle :\ G$ has a dominating set of size at most k $\}$
   a) Prove that both problems are in NP.
   b) Prove that VERTEX-COVER $\leq_P$ DOMINATING-SET

3. Say that two Boolean formulas on the same set of variables are ***output equivalent*** if they are true on exactly the same set of assignments to those variables. The ***length*** of a Boolean formula is the number of literals (occurrences of a variable negated or unnegated) it contains. For instance, the formula $(x_1 \vee \overline{x_2}) \wedge (x_1 \vee x_3 \vee x_2)$ has length 5. A Boolean formula is ***frugal*** if no shorter Boolean formula is equivalent to it.

   Consider the language FRUGAL-FORMULA $= \{\langle\phi\rangle : \phi$ is a frugal Boolean formula$\}$.
   Show that if $P = NP$ then FRUGAL-FORMULA $\in P$.

   [Here's a hint to get you started: Let EQU be the language of pairs of Boolean formulae $\langle\phi, \phi'\rangle$ such that ϕ and ϕ' are ouput-equivalent. Let NOT-EQU be the language of pairs of $\langle\phi, \phi'\rangle$ that are not equivalent. Show that NOT-EQU is in NP. So if $P = NP$ then NOT-EQU $\in P$. So EQU $\in P$ (why?). Now think about FRUGAL-FORMULA...]

OPTIONAL PROBLEM (You may do this problem for extra credit, OR you can do it instead of the first three problems!)

Recall the Graph Isomorphism problem: Two graphs H and G are isomorphic if the nodes of H can be reordered so it is identical to G.

Show that if $P = NP$ then the following problem has a polynomial-time solution: Given input graphs H and G that are isomorphic, **find the reordering of H that yields G.**

Note that the P and NP involve decision problems (with yes/no answers). But your algorithm should actually output the correct reordering!