Algorithms – CS-37000 Homework 9 – Feb 25, 2006

Instructor: László Babai Ry-164 e-mail: laci@cs.uchicago.edu

ADVICE. Take advantage of the TA's problem sessions. This is the **principal venue to discuss past homework and test problems.** Note that such problems may prop up in future tests.

READING. Review all previous handouts and readings. New reading, due Tuesday, Feb 28: text 8.1 (polynomial time reductions)

New reading, due Tuesday, March 7: text 6.3 (Segmented Least Squares (dynamic programming)); text pp.83–94 (depth-first seach (DFS)); 3.6 topological ordering pp.99-104.

HOMEWORK. Please **print your name on each sheet.** Put each solution on a separate sheet. Please try to make your solutions easily readable.

This homework is due on **Tuesday, February 28**.

9.1 (10 points) Suppose we have a black box that takes as input a pair (G, k) where G is an undirected graph and k is an integer, and outputs "YES" if G has a clique (complete subgraph) with k vertices. Given an undirected graph H, use this black box to find a largest clique in H in polynomial time, including a polynomial number of queries to the black box. *Hint.* First find the size of the largest clique; then, if this size is k, find a clique of size k.

For the next sequence of problems, we need the concept of *Karp-reductions* between languages.

Definition. Let $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$ be languages. We say that a function $f : \Sigma_1^* \to \Sigma_2^*$ is a **Karp-reduction** from L_1 to L_2 if

(i) f is polynomial-time computable;

(ii) $(\forall x \in \Sigma_1^*)(x \in L_1 \Leftrightarrow f(x) \in L_2)$.

We say that L_1 is **Karp-reducible** to L_2 (denoted by $L_1 <_K L_2$) if there exists a Karp-reduction from L_1 to L_2.

9.2 (10 points) Prove: if $L_2 \in P$ and $L_1 <_K L_2$ then $L_1 \in P$. Here P is the class of languages recognizable in polynomial time; recognizing a language means solving the membership problem for the language, i.e., deciding, for input $x \in \Sigma^*$, whether or not $x \in L$. Be specific about the exponents involved: if L_2 can be recognized in $O(n^a)$ and the function f can be computed in $O(n^b)$ then what is the smallest exponent c you can guarantee so that L_1 is recognizable in $O(n^c)$ where for each computation, n refers to the length of the input of that computation.

9.3 A linear inequality in the variables x, y, z is an inequality of the form $3x - 5y + z \leq 6$. (We may replace the coefficients $3, -5, 1, 6$ by arbitrary real numbers, and the number of variables may also be arbitrary. We shall assume, however, that *all coefficients are integers.*) A linear program (LP) is a set of linear inequalities with integer coefficients (including the right hand side). A LP is *feasible* if it has a solution (an assignment of real numbers to the variables satifying all inequalities).

Let LPf (*linear program feasibility*) denote the set of feasible LPs. Let ILPf (*integer linear program feasibility*) denote the set of those LPs which have an integer solution (all unknowns take integer values). Let $(0,1)$-ILPf denote the set of those LPs which have a $(0,1)$-solution (each unknown takes the value zero or one).

(i) (2 points) Give an example of a LP which is feasible (has solution(s) in real numbers) but which is not feasible as an ILP (has no integer solutions). Use as few variables as possible.

(ii) (4 points) Same as item (i) but all coefficients in the LP must be 0, 1, or -1, including the numbers on the right hand side. (A correct solution to this question also earns you the 2 points for part (i).)

(iii) (10 points) Let 3-COL denote the set (language) of 3-colorable graphs. Prove: 3-COL $<_K$ (0,1)-ILPf.