Purdue University — Hambrusch
CS 580 — Spring 2010

Assignment 8

Due: Tuesday, April 27, 2010 (before class)

FINAL EXAM: Tuesday, May 4, 8-10 am, KRAN G018

Qual Supplemental Exam: Friday, May 7, 9-10am, BRNG B222

1) (25 pts.)

(i) Describe a polynomial-time algorithm to determine if a graph has a clique of size 4. Does the existence of such an algorithm contradict the fact that the clique problem is NP-complete?

(ii) Show that if the decision version of the Clique problem can be solved in polynomial time, then there exists a polynomial-time algorithm for the optimization version of the Clique problem (return size of clique and vertices forming it).

(iii) If the decision version of the Clique problem can be solved in polynomial time, does there exist a polynomial-time algorithm that takes n cities and the distances between them and finds a TSP tour of minimum length through all cities.

(iv) You are supposed to design an efficient algorithm for a resource allocation (RA) problem on an undirected graph. A fellow student shows RA $<_{poly}$ Vertex Cover and states: "I have shown that RA is an NP-complete problem. Hence the most efficient algorithm I am likely to find has exponential time." You disagree. Who is correct and why?

(v) Another fellow student heard that problems are often easier on bipartite graphs (i.e., graphs in which every cycle has even length). The student claims to have found a reduction showing Hamiltonian cycle on undirected graphs G reduces in polynomial time to Hamiltonian cycle in bipartite graphs. The construction goes as follows: replace every edge (u, v) in G by a new vertex n_{uv} and two edges, (u, n_{uv}) and (n_{uv}, v). The resulting graph is clearly bipartite. The student claims it is easy to see that this is a correct transformation. Is there anything wrong with this transformation?

(vi) Which of the two problems on an undirected, n-vertex, m-edges graph $G = (V, E)$ in solvable in polynomial time and which is NP-complete? Explain your answers.

1. Given is a set L, $L \subseteq V$. Find a spanning tree T such that T's set of leaves includes the vertices in L.
2. Given is a set L, $L \subseteq V$. Find a spanning tree T such that T's set of leaves is precisely set L.

2) (13 pts.) In the Split-Equal (SE) problem we are given $2n$ positive integers, $a_1, a_2, \ldots, a_{2n}$, and are to determine whether it is possible to choose exactly n elements whose sum is equal to the sum of the remaining n elements. Show that the SE problem is NP-complete.

3) (12 pts.) Given are two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ and a positive integer k. The SG problem asks to determine whether there exists a graph G_3 such that G_3 has at least k edges and it is a subgraph of both G_1 and G_2. Show that the SG problem is NP-complete. (For the definition of a subgraph see Appendix B.4.)