CMPS 201 Seventhh Homework, Fall 04

3 problems, 30 pts, Due Tuesday, November 23

Reading: Chapter 34, and scan Chapter 35.

1. (13 pts) For this problem your are to first describe a language related to the 0-1 Knapsack optimization problem (see page 382 of the text) and show various things about your language.

   a. (1 pt) Fully describe the syntax of your language.

   b. (2 pts) show how a polynomial time algorithm which decides whether or not a string is in your language can be used as a subroutine to create a polynomial time algorithm solving the 0-1 Knapsack optimization problem.

   c. (2 pts) Show that your language is in NP.

   d. (8 pts) Complete the proof that your language is NP-complete by reducing a known NP-complete language to your language. Be sure to justify that your reduction is a polynomial time reduction and that it has the "if-and-only-if property". (Hint: you might want to choose SUBSET SUM as your known NP-complete problem.)

2. (12 pts) Given an undirected graph $G = (V, E)$, a *dominating set* in the graph is any set of vertices $V' \subseteq V$ such that each $v \in V - V'$ is adjacent to some vertex in V'. Show that the language $\{\langle G, k\rangle : G$ has a dominating set of size $k\}$ is NP-complete. (Hint: do a transformation from vertex cover which transforms each edge into a simple gadget. Make sure you justify that your reduction can be done in polynomial time and that it has the if-and-only-if property.)

3. (5 pts) Show that any language in NP can be decided by a deterministic algorithm running in time $2^{O(n^k)}$. The constant k may depend on the language in NP. A formal proof is not required.

Recommended problems: Exercise 34.1-3 and 34.1-6 on page 978-979, exercise 34.2-3 on page 983, exercise 34.3-6 and 34.3-7 on page 994, exercise 35.1-1 on page 1027.