Introduction to Computability and Complexity

Homework 1

1. P.64 of textbook: (a) minimum cover, (b) hitting set, (c) bounded degree spanning tree.

2. P.70 of textbook: (a) sequencing within intervals.

3. P.73 of textbook: (a) minimum tardiness sequencing.

4. P.75 of textbook: (a) longest path, (b) set packing, (c) largest common subgraph, (d) minimum sum of squares, (e) exact cover by 4-sets.

5. Suppose we have a set of n jobs to be scheduled on a single processor. Each job j has a processing time p_j, a weight w_j and a due date d_j. All jobs are available for processing at time 0. With respect to a schedule, a job is said to be on-time if it completes by its due date; otherwise, it is said to be late. The goal is to find a schedule that maximizes the total weight of all the on-time jobs. The decision version of this problem is to decide if there is a schedule such that the total weight of all the on-time jobs is at least K, where K is an additional parameter. Show that 0-1 knapsack is reducible to this decision problem.

6. Suppose we have $m \geq 1$ identical and parallel processors and a set of n tasks, $\{T_1, T_2, ..., T_n\}$. There is a directed acyclic graph defined on the set of tasks, in that if there is a directed edge from T_i to T_j, then T_j cannot start until T_i has finished execution. Each task has 1 unit of execution time. Given a time bound B, the question we ask: is there a schedule of the set of n tasks on the m processors such that the precedence constraint is observed and the makespan (schedule length) is no more than B?