# Spring 2013 CS 401 Homework 4

State all necessary assumptions clearly.
Due date: April 30, 2013 in class

1. Chapter 7, Exercise 3
2. Chapter 8, Exercise 2
3. Chapter 8, Exercise 5
4. In Chapter 4 we efficiently solved the Interval Scheduling problem using a greedy algorithm. We now consider a variant of this problem, known as Interval Scheduling with Windows.

   As before, you have a processor that is available to run jobs over some period of time (e.g., 6AM to 8PM). People submit jobs to run on the processor; the processor can only work on one job at any single point in time. Each job requires a set of intervals of time during which it needs to use the processor. Thus, for example, a single job could require the processor from 9AM to 10:15AM, and again from 5 PM to 6PM. If you accept this job, it ties up your processor during those specified intervals, but you could still accept jobs that need any other time periods (including from 10:15AM to 5PM).

   Given a set of $n$ jobs, each specified by a set of time intervals, you need to answer the question: For a given number $k$, is it possible to accept at least $k$ of the jobs so that no two of the accepted jobs have any overlap in time?

   Show that Interval Scheduling with Windows is NP-complete.

5. Consider an instance of the Satisfiability Problem, specified by clauses $C_1; ::: ; C_k$ over a set of Boolean variables $x_1; ::: ; x_n$. We say that the instance is monotone if each term in each clause consists of a nonnegated variable; that is, each term is equal to $x_i$, for some $i$, rather than $\overline{x_i}$. Monotone instances of Satisfiability are easy to solve: They are always satisfiable, by setting each variable equal to 1. For example, suppose we have the three clauses $(x_1 \vee x_2), (x_1 \vee x_3)$, $(x_2 \vee x_3)$. This is monotone, and indeed the assignment that sets all three variables to 1 satisfies all the clauses. But we can observe that this is not the only satisfying assignment; we could also have set $x_1$ and $x_2$ to 1, and $x_3$ to 0. Indeed, for any monotone instance, it is natural to ask how few variables we need to set to 1 in order to satisfy it. Given a monotone instance of Satisfiability, together with a number $k$, the problem of Monotone Satisfiability with Few True Variables asks: Is there a satisfying assignment for the instance in which at most $k$ variables are set to 1? Prove this problem is NP-complete.

6. Prove that Independent Set $\leq_p$ Set Packing.