CS:5350 Homework 10, Spring 2016

Due in class on Th, May 5

Collaboration: You are welcome to form groups of size 2 and work on your homeworks in groups. Of course, you are not required to work in groups. Every group should make one submission and names of both group members should appear on the submission and both students in a group will receive the same score. Other than the TA and the professor, you can only discuss homework problems with your group partner. Collaboration can be positive because talking to someone else about these problems can help to clarify your ideas and you will also (hopefully) get to hear about different ways of thinking about the problem. On the other hand, collaboration can be negative if one member of the group rides on work being done by the other member – please avoid this situation. If your solutions are (even partly) based on material other than what has been posted on the course website, you should clearly acknowledge your outside sources.
Late submissions: No late submissions are permitted. You will receive no points for your submission if your submission is not turned in at the beginning of class on the due date.
Evaluation: Your submissions will be evaluated on correctness *and* clarity. Correctness is of course crucial, but how clearly you communicate your ideas is also quite important.

1. The Partition problem is the following:
   Partition
   Input: A set S of integers.
   Question: Are there subsets A and B such that $A \cup B = S$ and $A \cap B = \emptyset$ and
   $$\sum_{a \in A} a = \sum_{b \in B} b?$$
   (a) Describe and analyze and algorithm to solve Partition in time $O(n \cdot M)$, where n is the size of the input set and M is the sum of the absolute values of its elements. (Think *dynamic programming*!)
   (b) In Prof. Erickson's notes, it is proved that Partition is NP-complete. Why doesn't this fact along with the algorithm in part (a) imply that $P = NP$?

2. Problem 2 from Chapter 30 in Prof. Jeff Erickson's notes (on "NP-hard Problems").

3. Recall that a set of vertices D in a graph $G = (V, E)$ is a *dominating set* if every vertex in G is either in D or has a neighbor in D. The decision version of the *minimum dominating set (MDS)* problem is as follows.
   MDS-DEC
   Input: A graph $G = (V, E)$ and a positive integer k.
   Output: Does G has a dominating set of size at most k?

   Show that MDS-DEC is NP-complete.
   Hint: To show NP-hardness of MDS-DEC use a polynomial reduction from MVC-DEC. Specifically, if $G = (V, E)$ is the graph that is part of the input to MVC-DEC, construct a graph G' as part of the input to MDS-DEC, by adding $|E|$ vertices and $2 \cdot |E|$ edges to G. How exactly to add these edges and vertices is for you to figure out.

4. Problem 28 from Chapter 8 "NP and Computational Tractability" from the textbook by Tardos and Kleinberg.
