CS 573: Graduate Algorithms, Fall 2011
HW 1 (due Tuesday, September 13th)

This homework contains five problems. **Read the instructions for submitting homework on the course webpage**. In particular, *make sure* that you write the solutions for the problems on separate sheets of paper. Write your name and netid on each sheet.

Collaboration Policy: For this home work students can work in groups of up to three students each. Only one copy of the homework is to be submitted for each group. Make sure to list all the names/netids clearly on each page.

Note on Proofs: Details are important in proofs but so is conciseness. Striking a good balance between them is a skill that is very useful to develop, especially at the graduate level.

1. (20 pts) Proofs are not necessary for the following two problems which should be easy reductions from known problems.

   - (10 pts) k-Bounded Set Cover is the special case of Set Cover in which the cardinality of each set is at most k. Describe a reduction to show that 3-Bounded Set Cover is NP-Hard.
   - (10 pts) A directed graph $G = (V, E)$ is *strongly connected* if for all $u, v \in V$ there is a path in G from u to v and from v to u. The Strongly-Connected Spanning Subgraph problem is the following: given a directed graph $G = (V, E)$ and an integer k is there a spanning subgraph H of G with at most k edges such that H is strongly connected? A subgraph is spanning if it contains all the vertices of the original graph. Describe a reduction to show that Strongly-Connected Spanning Subgraph is NP-Hard.

2. (20 pts) Given a graph $G = (V, E)$ a subset $S \subseteq V$ of nodes is called a *dominating set* if for each node $v \in V$ the node $v \in S$ or there is an edge $uv \in E$ such that $u \in S$. The Dominating Set problem is an optimization problem in which the goal is to find a smallest dominating set in a given graph (note that you want to find a set and not just the size). In the decision version we are given G and an integer k and the goal is to check whether G has a dominating set of size at most k. A problem is said to be *self-reducible* if its optimization version can be reduced in polynomial time to its decision version. Show that Dominating Set is self-reducible. Formally, given a black box access to a subroutine that solves the decision version of Dominating Set give a polynomial time algorithm that solves the optimization version of Dominating Set. *Note:* Dominating Set can be shown to be NP-Hard via a relatively easy reduction from Set Cover. You do no need to prove this though.

3. (20 pts) Pos-or-Neg 3-SAT is a variant of 3-SAT in which the given 3-SAT formula has the property that each cluase has all of the literals as positive variables or all of the literals as negations of the variables. Prove that Pos-or-Neg 3-SAT is NP-Hard.

4. (20 pts) In the Subset Sum problem the input consists of n integers $a_1, a_2, \ldots, a_n$ and another integer B and the goal is to check if there is a subset of the a_i's that exactly sum

to B. Consider the PARTITION problem where the input consists of n integers $a_1, \ldots, a_n$ and the goal is to check whether the numbers can be partitioned into two sets S_1 and S_2 such that the sum of the numbers in S_1 is exactly equal to that of S_2.

- (15 pts) Show that PARTITION is NP-Hard.
- (5 pts) Consider the SQUARED SUM PARTITION problem where the input consists of n integers $a_1, \ldots, a_n$ and integers k, B. The goal is to check if there a partition of the given numbers into k sets $S_1, \ldots, S_k$ such that

$$\sum_{i=1}^{k} \left(\sum_{a_j \in S_i} a_j \right)^2 \leq B.$$

Show that SQUARED SUM PARTITION is NP-Hard. No proof necessary for this part.

5. (20 pts) Problem 8.23 in Chapter 8 of Dasgupta etal book that is available here. `http://www.cs.berkeley.edu/~vazirani/algorithms.html`.

6. (Extra Credit: 20pts) Given a graph $G = (V, E)$ a matching M is a set of edges such that the degree of each node in graph induced by M is at most one. We say that a set of edges M' is an *or-matching* if for each edge $uv \in M'$ at least one of u and v has degree one in the graph induced by M'. The OR-MATCHING problem is the following: given a graph $G = (V, E)$ and an integer k is there an or-matching M' in G with at least k edges? Show that the OR-MATCHING problem is NP-Hard.

Questions to ponder: I am listing below a few questions that you may want to think about to further your understanding.

1. Show that $PSPACE \subseteq EXP$.

2. Derive a polynomial-time algorithm for 2-SAT. See Problem 3.28 in Dasgupta etal book.

3. Think of a problem or read up on a problem that is decidable but is not known to be in NP.

4. The BIN-PACKING problem is the following. Given n items with rational-valued sizes $a_1, \ldots, a_n \in [0, 1]$ and an integer m can the items be packed in m bins of size 1 each? Show that Bin Packing is *strongly* NP-Complete via a reduction from 3-Partition. If m is a fixed constant can you come up with a *pseudo-polynomial* time algorithm?