Homework 10 – Due Friday, December 3, 2010

Please refer to the general information handout for the full homework policy and options.

Reminders

- Your solutions are due before the lecture. Late homework will not be accepted.

- Collaboration is permitted, but you must write the solutions by yourself without assistance, and be ready to explain them orally to a member of the course staff if asked. You must also identify your collaborators. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

- To facilitate grading, please write down your solution to each problem on a separate sheet of paper. Make sure to include all identifying information and your collaborators on each sheet.

- For problems that require you to provide an algorithm, you must give a precise description of the algorithm, together with a proof of correctness and an analysis of its running time. You may use algorithms from class as subroutines. You may also use any facts that we proved in class or from the book.

Reminder  To show that a problem is NP-complete, you must show both that it is in NP and that it is NP-hard. The easiest way to show that problem $X$ is NP-hard is to prove $Y \leq_{p,Karp} X$ (“there is a Karp reduction from $Y$ to $X$” or “$Y$ Karp-reduces to $X$”), where $Y$ is an NP-hard problem already covered in class.

Exercises  You do not need to hand these in.

1. Consider the following variant of the Diverse Subset problem (from KT, Chapter 8, Problem 2): Suppose the store sells internet access by the hour. The products are now minutes, and each customer buys a contiguous interval of time. Is the problem still NP-hard?

2. (Search to Decision Reduction for SAT) Let SAT be the decision problem defined on page 459 of KT. Let SAT-SEARCH be the search version of the problem, where the input is a formula $\Psi$ and the goal is to output a satisfying assignment for $\Psi$ if one exists. Show that

   $$\text{SAT-SEARCH} \leq_{p,Cook} \text{SAT}.$$  

   In other words, show how to solve SAT-SEARCH is polynomial time, given an oracle for SAT. (Hint: Figure out a good assignment for one variable at a time.) Analyze the running time of your algorithm, its space complexity and the number of calls to the oracle.

   [For extra fun, do the same exercise with any NP-complete problem. Almost all NP-complete problems admit a simple reduction from the search version to the decision version. You have to do this for credit for the 3D matching problem.]
3. Suppose you are running a web site that is visited by the same set of people every day. Each visitor claims membership in one or more demographic groups; for example, a visitor might describe himself as male, 31-40 years old, a resident of Pennsylvania, an academic, a father of two and a lenient grader. Your site is supported by advertisers. Each advertiser has told you which demographic groups should see its ads and how many of its ads you must show each day. Altogether, there are \( n \) visitors, \( k \) demographic groups, and \( m \) advertisers.

Let \textsc{demogadery} be the problem of determining, given all the data described in the previous paragraph, whether you can show each visitor exactly one ad per day, so that every advertiser has its desired number of ads displayed, and every ad is seen by someone in an appropriate demographic group.

Prove either that \textsc{demogadery} is in P or that it is NP-complete.

**Problems to be handed in**

1. KT book, Chapter 8, problem 8 (Madison’s letters). You can reduce from 3-dimensional matching (although a reduction from any NP-complete problem discussed in the book will do).

2. Consider the degree-\( k \) spanning tree problem: given an undirected graph \( G = (V, E) \), the goal is to find a spanning tree \( T \) of \( G \) such that the degree of every node in \( T \) is at most \( k \) (that is, every node is incident to at most \( k \) edges in \( T \)).

   (a) What problem discussed in Chapter 8 of the textbook is equivalent to the degree-2 spanning tree problem?

   (b) Show that, for every integer \( k > 1 \), the degree-\( k \) spanning tree problem is NP-complete.  
   \([Hint: \text{Give a reduction from the degree-2 spanning tree problem to the degree-}\( k \) spanning tree problem.}\]

3. We saw in class that 3-SAT is NP-complete. Consider instead 2-SAT, in which the input is a formula with at most 2 literals per clause. Show that 2-SAT is in P.

4. (**Search to decision reduction for 3-D matching**) Your fairy godmother has given you a magic box that can solve the decision version of the 3-D matching problem in constant time. (That is, given an arbitrary instance of 3-D matching, it will quickly report whether a three-way matching exists, but it tells you nothing about which triples are used in the matching). Give an algorithm that uses this box as a subroutine to find and output a 3-D matching in polynomial time, if one exists. Explain carefully why your algorithm is correct and analyze its space and time complexity. Remember that you need time to write down the instances on which you will be using your magic box.

5. Short programming problems, to be discussed in a separate handout (different due date).