

## Homework 11 – Due Friday, December 5, 2008

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. Your solutions to different problems will be graded separately, possibly by different people, and returned to you independently of each other.
- 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  $Y \leq_P X$  where  $Y$  is an NP-hard problem already covered in class.

**Exercises** These should not be handed in, but the material they cover may appear on exams:

1. KT Chapter 8, Problems 1–6, 8, 13.

### Problems to be handed in

<p><b>Page limits:</b> The answer to each problem should fit in <b>2 pages (or one double-sided sheet)</b> of paper. Longer answers will be penalized.</p>
--

1. (**Resource Reservation Problem**) KT, Chapter 8, Problem 4.  
(It may help to also look at KT, problem 2.)

(See over.)

2. (**Search vs. Decision Problems**) 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

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

In other words, show how to solve SAT-SEARCH in 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. It may help to review the self-reduction for vertex-cover seen in class.)