Homework 10 – Due Wednesday, November 28, 2007

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 all problems where you are asked to design an algorithm, do not forget to prove correctness and analyze your algorithm's time and space complexity.*

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

1. (**Office Hours**) Chapter 7, Problem 28.

Problems to be handed in

1. (**Moving Applications**) Chapter 7, Problem 29.
2. (*k* **Edge-Disjoint Paths**) Chapter 7, problem 32.
3. (**Diverse Subset Problem**) Define DIVERSE-SUBSET problem as in Problem 2, Chapter 8. Give a Karp reduction from DIVERSE-SUBSET to one of the problems we defined in lecture 35. In other words, give a polynomial time transformation that, given an instance of DIVERSE-SUBSET produces an instance of the problem you are reducing to. YES instances should be transformed into YES instances, and NO instances – into NO instances. Prove that your reduction is correct (satisfies the requirement in the previous sentence), and analyze its running time and space complexity.
4. (**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 Ψ and the goal is to output a satisfying assignment for Ψ 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.