Computability and Complexity — Winter 2009

Problem Set #4 — Instructor: Anna Karlin

Due on **February 13, 2009** by email to `ncthach@cs.washington.edu`

Instructions: You are allowed to discuss the problems with fellow students taking the class. However, you must write up your solutions completely on your own. Moreover, if you do discuss the problems with someone else, I am asking, on your honor, that you do not take any written material away from the discussion. In addition, for each problem on the homework, I ask that you acknowledge the people you discussed that problem with, if any.

The problems have been carefully chosen for their pedagogical value and hence might be similar or identical to those given out in past offerings of this course at UW, or similar courses at other schools. Using any pre-existing solutions from these sources, from the Web or other textbooks constitutes a violation of the academic integrity expected of you and is strictly prohibited.

Most of the problems require only one or two key ideas for their solution – spelling out these ideas should give you most of the credit for the problem even if you err in some finer details. So, make sure you clearly write down the main idea(s) behind your solution.

A final piece of advice: Begin work on the problem set early and don't wait till the deadline is a day or two away.

Readings: Arora/Barak Sections 4.1-4.2, Sipser Sections 8.1 – 8.3

1. Let *DOUBLE-SAT* be the set of CNF formulas ϕ such that ϕ has at least two satisfying assignments. Show that *DOUBLE-SAT* is NP-complete.

2. The k-*SPANNING-TREE* problem is the following. Given an undirected graph $G = (V, E)$, determine if there is a spanning tree of G in which each node has degree at most k.

   Show that for any $k \geq 2$, k-*SPANNING-TREE* is NP-complete. (Hint: start with $k = 2$.)

3. This problem is inspired by the single-player game *Minesweeper*, generalized to an arbitrary graph. Let G be an undirected graph, where each node either contains a single, hidden *mine* or is empty. The player chooses nodes, one by one. If the player chooses an empty node, the player learns the number of neighboring nodes containing mines. (A neighboring node is one connected to the chosen node by an edge.) The player wins if and when all empty nodes have been so chosen.

   In the *mine consistency problem* you are given a graph G, along with numbers labeling some of G's nodes. You must determine whether a placement of mines on the remaining nodes is possible, so that any node v that is labeled m has exactly m neighboring nodes containing mines. Formulate this problem as a language and show that it is NP-complete.

4. *Extra Credit:* There are many different ways to formalize the intuitive problem of *clustering*, where the goal is to divide up a collection of objects into groups that are "similar" to each other.

   Here we want to formalize the clustering problem as follows. Divide the objects up into k sets so as to *minimize* the maximum distance between any pair of objects in the *same* cluster.

Formally, we are given n objects $p_1, p_2, \ldots, p_n$ with a numerical distance $d(p_i, p_j)$ defined on each pair of objects. (We require that $d(p_i, p_i) = 0$; that $d(p_i, p_j) > 0$ if $p_i \neq p_j$ and that distances are symmetric $d(p_i, p_j) = d(p_j, p_i)$.) We are also given an integer k and a bound B. The *Low-Diameter-Clustering-Problem* is the following: Can the objects be partitioned into k sets, so that no two points in the same set are at a distance greater than B from each other?

Prove that the *Low-Diameter-Clustering-Problem* is NP-complete.

5. *Extra Credit:* We now consider a different clustering problem, in which we try to *maximize* the distance between clusters. Formally, we are given n objects $p_1, p_2, \ldots, p_n$ with a numerical distance $d(p_i, p_j)$ defined on each pair of objects as above, an integer k and a value B. The *Large-InterCluster-Distance-Problem* is the following: Can the objects be partitioned into k sets, so that the minimum distance between two points in *different* sets is at least B?

   Prove that the *Large-InerCluster-Distance-Problem* is in P.