# 22C:31 Homework 2

## Due in class on Tuesday, Feb 23rd

This homework will be graded out of 60 points and it is worth 6% of your grade. The Teaching Assistant will grade some 6 out of the 8 problems, with each problem worth 10 points.

1. Problem 3, Chapter 2, Page 67.
2. Problem 4, Chapter 2, Pages 67-68.
3. Solve the Base Station Placement problem, i.e., Problem 5, Chapter 4, Page 190. In addition to describing the algorithm in pseudocode, present a proof of correctness as well. For the greedy algorithm I have in mind, a simple "greedy stays ahead" type of proof suffices.
4. Consider this 2-dimensional version of the Base Station Placement problem. You are given a set $P$ of $n$ points in the plane. You can locate a cell phone base station at each point $p \in P$. A base station located at $p$ is said to *cover* all points in $P$ that are within 1 unit of distance from $p$. This basically means that the transmission range of each base station is a unit disk (i.e., disk of radius one) centered at that base station.
   (a) State a greedy algorithm for solving the problem of finding a minimum set of base stations that cover all points in $P$. Whatever greedy algorithm you state should "pass" the basic sanity check that if there is an optimal solution to the problem with one base station, then your algorithm should find it.
   (b) Find a simple counterexample for this greedy algorithm.
5. Problem 13, Chapter 4, Pages 194-195. No proof of correctness required.
6. Problem 16, Chapter 4, Pages 196-197. No proof of correctness required.
7. Alice wants to throw a party and is deciding whom to invite. She has $n$ people to choose from, and she has made up a list of which pairs of these people know each other. She wants to pick *as many people as possible*, subject to two constraints: at the party each person should have at least five other people they know and at least five other people whom they don't know.

   Give an efficient algorithm that takes as input the list of $n$ people and the list of pairs who know each other and outputs the best choice of party invitees. Give the running time in terms of $n$.
8. The *0-1 knapsack problem* is posed as follows. A thief robbing a store finds $n$ items; the $i$th item is worth $v_i$ dollars and weighs $w_i$ pounds, where $v_i$ and $w_i$ are integers. He wants to take as valuable a load as possible, but he can carry at most $W$ pounds in his knapsack for some integer $W$. Which items should he take?
   (a) A simple greedy algorithm for the problem would pick (from all available items) an item $i$ with maximum value per pound (i.e., $v_i/w_i$ is maximum). This item is unavailable for further consideration and is added to the solution provided adding it still satisfies the knapsack weight constraint. This continues until Devise a simple counterexample to the claim that this greedy algorithm always produces an optimal solution to the 0-1 knapsack problem.

(b) In the *fractional knapsack problem*, the setup is the same, but the thief can take fractions of items, rather than make a binary (0-1) choice for each item. The greedy algorithm described above for the 0-1 knapsack problem has a version that works correctly for the fractional knapsack problem. Present the algorithm (in pseudocode) and present its proof of correctness.

---