CMPS 201
Spring 2010
Homework Assignment 7

1. (1 Point) What is the worst-case running time of the Bucket Sort algorithm (on page 174, 2nd edition)? What simple change to the algorithm preserves its linear average case running time, and makes its worst-case running time $\Theta(n\log n)$?

2. (1 Point) Let h be height of a k-ary tree having n leaves. Use induction on h to prove $h \ge \lceil \log_k n \rceil$. (Hint: follow the theorem on binary trees proved in class.)

3. (1 Point) Consider the problem of determining an integer m in the range $1 \le m \le 10^9$ by asking a sequence of questions which have at most 5 possible answers. (For instance, you may ask to which of 5 subintervals m belongs.) Give a decision tree lower bound on the worst case number of questions which must be asked by any algorithm that solves this problem.

4. (3 Points) **Bar Weighing Problem:** Assume we are given 12 gold bars numbered 1 to 12 where 11 bars are pure gold and one is counterfeit: either gold-plated lead (which is heavier than gold), or gold-plated tin (lighter than gold). The problem is to find the counterfeit bar and what metal it is made of using only a balance scale. Any number of bars can be placed on each side of the scale, and each use of the scale produces one of three outcomes: either the left side is heavier, or the two sides are the same weight, or the right side is heavier.

   a. (1 Point) Give a decision tree argument to establish a lower bound on the (worst case) number of weighings which must be performed by any algorithm which solves this problem.
   b. (1 Point) Design an algorithm which solves this problem with the (worst case) number of weighings equal to the lower bound you found in (a). Present your algorithm by drawing a decision tree, rather than pseudo-code.
   c. (1 Point) Alter the problem slightly so as to allow the possibility that all 12 bars are pure gold, and thus there is one additional possible verdict, i.e. "all gold". Make a minor change to your algorithm in part (b) so it gives a correct answer in this more general setting.

5. (2 Points) **Water Jug Problem** (problem 8-4 on page 179 of 2nd edition): Suppose that you are given n red and n blue water jugs, all of different shapes and sizes. All red jugs hold different amounts of water, as do the blue ones. Moreover, for every red jug, there is a blue jug that holds the same amount of water, and vice versa. It is your task to find a grouping of the jugs into pairs of red and blue jugs that hold the same volume of water. To do so, you may perform the following operation: pick a pair of jugs, one red, one blue, fill the red jug with water, and then pour the water into the blue jug. This operation will tell you if the two jugs hold the same amount of water, and if not, which one holds more water. Assume that such a comparison takes one unit of time. Your goal is to find an algorithm that solves this problem. Remember that you may not directly compare two red jugs or two blue jugs.

   a. (1 Point) Describe a simple algorithm that uses $\Theta(n^2)$ comparisons (in worst case) to group the jugs into pairs.
   b. (1 Point) Prove a lower bound of $\Omega(n\log n)$ for the worst case number of comparisons which any algorithm that solves this problem must perform.