CMPS 102
Homework Assignment 8

1. Let m be an integer in the set $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, and consider the following problem: determine m by asking 3-way questions, i.e. questions with at most 3 possible answers. For instance, one could ask which of 3 specific subsets m belongs to.

   a. Give a decision tree argument showing that at least 3 such questions are necessary in worst case. In other words, prove that no correct algorithm can solve this problem by asking only 2 questions in worst case.

   b. Design an algorithm that will solve this problem by asking 3 such questions in worst case. Express your algorithm as a decision tree.

2. **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. Give a decision tree lower bound for the (worst case) number of weighings that must be performed by any algorithm solving this problem.

   b. Design an algorithm that solves this problem with (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. Alter the problem slightly to allow the possibility that all 12 bars are pure gold. Thus there is one additional possible verdict: "all gold". Make a minor change to your algorithm in part (b) so that it gives a correct answer to this more general problem.

3. **Water Jug Problem** (Problem 8-4: page 206 of CLRS 3rd 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. Find an algorithm that uses $\Theta(n^2)$ comparisons (in worst case) to group the jugs into pairs.

b. Prove a lower bound of $\lceil \log_3(n!) \rceil$ for the worst case, and $\log_3(n!)$ for the average case number of comparisons to be performed by any algorithm that solves this problem.

c. Find an algorithm for this problem that runs in (average case) time $\Theta(n \log n)$. (Hint: modify the algorithms Partition() and Quicksort().)