CMPS 201
Homework Assignment 7

1. Recall the coin changing problem: Given denominations $d = (d_1, d_2, \dots, d_n)$ and an amount N to be paid, determine the number of coins in each denomination necessary to disburse N units using the fewest possible coins. Assume that there is an unlimited supply of coins in each denomination.

   a. Write pseudo-code for a greedy algorithm that attempts to solve this problem. (Recall that the greedy strategy doesn't necessarily produce an optimal solution to this problem. Whether it does or not depends on the denomination set d.) Your algorithm will take the array d as input and return an array G as output, where $G[i]$ is the number of coins of type i to be disbursed. Assume the denominations are arranged by increasing value $d_1 < d_2 < \cdots < d_n$, so your algorithm will step through array d in reverse order. Also assume that $d_1 = 1$ so any amount can be paid.
   b. Let $d_i = b^{i-1}$ for some integer $b > 1$, and $1 \le i \le n$, i.e. $d = (1, b, b^2, \dots, b^{n-1})$. Does the greedy strategy always produce an optimal solution in this case? Either prove that it does, or give a counter-example.
   c. Let $d_1 = 1$ and $2d_i \le d_{i+1}$ for $1 \le i \le n-1$. Does the greedy strategy always produce an optimal solution in this case? Either prove that it does, or give a counter- example.

2. **Activity Scheduling Problem:** Consider n activities $\{1, 2, \dots, n\}$ with start times $s_1, \dots, s_n$ and finish times $f_1, \dots, f_n$, that must use the same resource (such as lectures in a lecture hall, or jobs on a machine.) At any time only one activity can be scheduled. Two activities i and j are *compatible* if their time intervals $[s_i, f_i]$ and $[s_j, f_j]$ have non-overlapping interiors. Your objective is to determine a set of compatible activities of maximum possible size. For each of the greedy strategies below, determine whether or not it provides a correct solution to all instances of the problem. If your answer is yes, state and prove a theorem establishing the correctness of the proposed strategy. If your answer is no, provide a counterexample (i.e. specific start and end times) showing that the strategy can fail to find an optimal solution.

   a. Order the activities by increasing total duration. Schedule activities with the shortest duration first, satisfying the compatibility constraint. If there is a tie, choose the one that starts first.
   b. Order the activities by increasing start time. Schedule the activities with the earliest start times first, satisfying the compatibility constraint. If there is a tie, choose the one having shortest duration.
   c. Order the activities by increasing finish times. Schedule the activities with the earliest finish times first, satisfying the compatibility constraint. If there is a tie, pick one arbitrarily.

3. 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.

4. **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.

5. **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().)