CSE 102
Homework Assignment 6

1. (Read the **Rod-Cutting Problem** in section 15.1 pp. 360-369 of CLRS 3rd edition. This is problem 15.1-2 on p. 370.) Show, by means of a counterexample, that the following "greedy" strategy does not always determine an optimal way to cut rods. Define the *density* of a rod of length i to be p_i/i, that is, its value per inch. The greedy strategy for a rod of length n cuts off a first piece of length i, where $1 \le i \le n$, having maximum density. It then continues by applying the greedy strategy to the remaining piece of length $n - i$.

2. Recall the coin changing problem: Given denominations $d = (d_1, d_2, \ldots, d_n)$ 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. $d = (1, b, b^2, \ldots, b^{n-1})$. 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.

3. **Activity Scheduling Problem:** Consider n activities $\{1, 2, \ldots, n\}$ with start times $s_1, \ldots, s_n$ and finish times $f_1, \ldots, f_n$, 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.