CMPS 102
Homework Assignment 7
Solutions

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.

   Solution:
   <u>GreedyCoinChange(d, N)</u> (Pre: $1 = d[1] < d[2] < \cdots < d[n]$)

   ```
   n = length[d]
   G[1···n] = (0, 0, ..., 0)
   sum = 0
   i = n
   while sum < N
       while sum + d[i] ≤ N
           sum += d[i]
           G[i]++
       i--
   return G
   ```

   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.

   Theorem
   The greedy strategy produces an optimal solution with this denomination set for any amount N.

   Proof:
   Let $N \ge 0$, and suppose that $x = (x_1, x_2, \dots, x_n)$ is an optimal disbursement of N units, where x_i is the number of coins of type i to be paid ($1 \le i \le n$). Let $g = (g_1, g_2, \dots, g_n)$ be the solution produced by the greedy strategy. To show that g is optimal, it suffices to show that $g_i = x_i$ for $1 \le i \le n$. Since both solutions disburse N units, we have $\sum_{i=1}^{n} x_i b^{i-1} = N = \sum_{i=1}^{n} g_i b^{i-1}$, i.e.

   $$x_1 + x_2 b + x_3 b^2 + \cdots + x_n b^{n-1} = g_1 + g_2 b + g_3 b^2 + \cdots + g_n b^{n-1}.$$

   Reducing this equation mod b yields the congruence $x_1 \equiv g_1 (\operatorname{mod} b)$. Observe $0 \le x_1 < b$, since otherwise it would be possible to replace b coins of type 1 by 1 coin of type 2, contradicting that x is optimal. Likewise $0 \le g_1 < b$, since the greedy strategy guarantees that as many coins as possible of type 2 (value b) will be disbursed before any coins of type 1 are. These inequalities, along with the preceding congruence, imply $x_1 = g_1$. Cancel x_1 from both sides of the above equation and divide through by b to obtain

$$x_2 + x_3 b + \cdots + x_n b^{n-2} = g_2 + g_3 b + \cdots + g_n b^{n-2}.$$

A similar argument shows that $x_2 = g_2$. Continuing in this manner, we have that $g_i = x_i$ for $1 \le i \le n$, and hence the greedy solution g is in fact optimal, as required. ■

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.

Counter Example:
Let $n = 3$, $d = (1, 10, 25)$, and $N = 30$. Observe that $2 \cdot 1 \le 10$, and $2 \cdot 10 \le 25$. The greedy solution in this case is $g = (5, 0, 1)$, which uses 6 coins, while the optimal solution $x = (0, 3, 0)$ uses only 3 coins. ■

2. **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.

Counter-Example:

Activity:	1	2	3
Start time:	2	0	3
Finish time:	4	3	6

The greedy strategy yields the schedule {1}, while the optimal schedule is {2, 3}.

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.

Counter-Example:

Activity:	1	2	3
Start time:	0	1	2
Finish time:	3	2	4

The greedy strategy yields the schedule {1}, while the optimal schedule is {2, 3}.

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.

Theorem
This strategy maximizes the number of scheduled activities.

Proof:
Let $A = \{1, 2, \dots, n\}$ denote set of activities, let $s(1), s(2), \dots, s(n)$ be the start times, and let $f(1), f(2), \dots, f(n)$ be the finish times. Assume that A is already ordered by finish times: $f(1) \leq f(2) \leq \cdots \leq f(n)$. Observe two activities i and j with $i < j$ are compatible iff $f(i) \leq s(j)$. We call a subset of A *feasible* if all its activities are mutually compatible. We call a feasible set *optimal* if it has maximum cardinality amongst all feasible subsets of A.

Let $X = \{i_1 < i_2 < \cdots < i_k\} \subseteq A$ be the subset obtained by the greedy strategy described above, and let $Y = \{j_1 < j_2 < \cdots < j_l\} \subseteq A$ be any optimal subset, so in particular $l \geq k$. Clearly the greedy strategy guarantees X is feasible. We must show that X is also optimal, i.e. $l = k$.

If $X = Y$ there is nothing to prove, so suppose $X \neq Y$. Let t be the first index $1 \leq t \leq k$ such that $i_t \neq j_t$. In other words $i_r = j_r$ for all $r < t$, and $i_t \neq j_t$. (Such an index t must exist for otherwise $i_r = j_r$ for all $1 \leq r \leq k$, and hence $X \subseteq Y$. Since $X \neq Y$ there must be an element $j_{k+1} \in Y - X$ that is compatible with all the activities in X. But the greedy strategy should then have added j_{k+1} to X. Since it did not, an index t with the above properties must exist.) Both activities i_t and j_t must therefore be compatible with $\{i_1, i_2, \dots, i_{t-1}\} = \{j_1, j_2, \dots, j_{t-1}\}$. Since i_t was chosen by the greedy strategy we must have $f(i_t) \leq f(j_t)$. The compatibility constraint implies that $f(j_t) \leq s(j_{t+1})$, and therefore $f(i_t) \leq s(j_{t+1})$. It follows that i_t is also compatible with j_{t+1}, and hence also with all the activities in the set $\{j_{t+1}, \dots, j_l\}$.

Let $Y_1 = Y - \{j_t\} \cup \{i_t\}$. The above remarks show that Y_1 is a feasible set, and since $|Y_1| = |Y|$, Y_1 is optimal as well. But Y_1 has one more activity in common with X than Y does. If $X = Y_1$, then X is optimal and we are done. If not, we repeat the above process with Y_1 in place of Y, to obtain another optimal set Y_2 having one more activity in common with X than Y_1. Continuing in this fashion we construct a sequence of optimal sets, each having more in common with X than the previous. Eventually we must reach an optimal set Y_m that is identical to X, showing that X is itself optimal as claimed. ■