1. Prove or disprove the following properties of running time functions:

   (a) If $f(n) \in O(n)$, then $[f(n)]^2 \in O(n^2)$.

   (b) If $f(n) \in O(n)$, then $2^{f(n)} \in O(2^n)$.

2. Consider the following algorithm:

```
procedure waste(n):
    for i ranging from 1 to n, do
        for j ranging from 1 to i, do
            write i, j, n;
    if n > 0 then
        for i ranging from 1 to 4, do
            waste(n + 2).
```

   Let $T(n)$ stand for the number of lines of output generated by a call of waste(n). Provide a recurrence equation for $T(n)$, and determine the exact value of $T(n)$ or its order, in the simplest possible form.

3. Let m, n, z be positive integers.

   (a) Prove that $(m\ n) \bmod z \ = \ [(m \bmod z) \times (n \bmod z)] \bmod z$
   and $(m^n) \bmod z \ = \ [(m \bmod z)^n] \bmod z$.

   (b) How would you use these equations, if possible, to develop divide and conquer algorithms to evaluate $(m\ n) \bmod z$ and $(m^n) \bmod z$?

   (c) Provide recurrence equations for the running time for each of the divide and conquer algorithms you were able to construct in part (b). You do not have to solve the recurrence equation(s).

4. Answer the following questions for (each of) Kruskal's algorithm and Prim's algorithm to obtain the minimal spanning tree of a graph.

   (a) What happens if each algorithm is executed on a graph that is not connected?

   (b) What happens if each algorithm is executed on a graph some of whose edges have negative weights?

5. The problem of "making change" requires paying a specified amount to a customer using the smallest possible number of coins. Develop a dynamic programming solution to solve the "making change" problem, assuming that we have an unlimited supply of coins, the amount to be paid to the customer is N units, and that there are coins of n different denominations, with each coin's value $\in \{d_1,\ d_2,\ \ldots,\ d_n\}$ (where each d_i, n and N is a positive integer).

   For example, if we live in a country that has coins for 1, 4 and 6 units, and the specified amount to be paid to a customer is 8 units (i.e., $n = 3,\ d_1 = 1,\ d_2 = 4,\ d_3 = 6,\ N = 8$), then the optimum solution consists of two coins, both of value 4 units. The greedy algorithm that uses a 6-unit coin first, followed by two 1-unit coins, is not optimal.

6. (a) Discuss the relationship between inserting into a binomial hep and incrementing a binary number.

   (b) Discuss the relationship between uniting two binary heaps and adding two binary numbers.

   (c) Can your answer above be used to formulate an algorithm for uniting two binary heaps? Explain.

7. An algorithm to find the greatest common denominator (GCD) of positive integers $u,\ v$ is given below.

   $n := 0$;
   L1: $t := |u - v|$;
   if $t = 0$ then return u as the GCD and halt;
   L2: $t := t/2$;
   if t is even then go to L2;
   L3: $n := n - 1$;
   if $u \geq v$ then $u := t$ else $v := t$;
   go to L1.

   Let one "iteration" refer to one execution of step L3 in the algorithm.

   (a) Why can we assume, without loss of generality, that u and v are odd?

   (b) Use this algorithm to find GCD(395, 275).

   (c) How many iterations were performed to find GCD(395, 275)?

   (d) How many iterations will be required to find GCD(u,v) for arbitrary u, v? Comment on the worst case behavior of the algorithm.