CSE 321: Discrete Structures
Assignment #5
due: Wednesday, Nov 5

1. Euclids algorithm for computing the greatest common divisor of two numbers $a > b \geq 0$ can be written recursively as follows:

   ```
   procedure Euclid ($a, b$: integer) returns $g$: integer
   begin
       if $b = 0$ then return $a$;
       else return
           Euclid ($b$, $a$ mod $b$);
   end .
   ```

   Prove by induction that for all $k \geq 2$, if Euclid(a, b) performs at least k recursive calls, then $a \geq f_{k+1}$ and $b \geq f_k$, where f_i is the i-th Fibonacci number.

2. It is a somewhat amazing fact that the greatest common divisor can be written as a linear combination, that is, $\gcd(a, b) = sa + tb$, for some integers s and t. It is sometimes important to be able to compute not only the greatest common divisor, but the coefficients s and t as well. (Part (b) of this problem gives an example application.) The following extension of Euclid's algorithm computes the gcd g plus those coefficients. Try it out on some examples.

   (The programming notation $(x, y) \leftarrow (e, f)$ means simultaneous assignments of the old value of e to x and the old value of f to y. For instance, the body of the ordinary Euclidean algorithm's loop could have been written $(x, y) \leftarrow (y, x \bmod y)$. Note that this is exactly the effect of the statement $(a_0, a_1) \leftarrow (a_1, a_0 - q * a_1)$ below, so that the output g is still $\gcd(a, b)$.)

   ```
   procedure Extended_Euclid ($a, b$: integer) returns $g, s, t$: integer
   begin
       $(a_0, a_1) \leftarrow (a, b)$;
       $(s_0, s_1) \leftarrow (1, 0)$;
       $(t_0, t_1) \leftarrow (0, 1)$;
       while $a_1 \neq 0$ do
       begin
           $q \leftarrow \lfloor a_0/a_1 \rfloor$;
           $(a_0, a_1) \leftarrow (a_1, a_0 - q * a_1)$;
           $(s_0, s_1) \leftarrow (s_1, s_0 - q * s_1)$;
           $(t_0, t_1) \leftarrow (t_1, t_0 - q * t_1)$;
       end ;
       $g \leftarrow a_0$;
       $s \leftarrow s_0$;
   ```

```
    $t \leftarrow t_0$;
**end** .
```

(a) Prove that the inputs and outputs satisfy $g = sa + tb$. (Hint: Use induction to prove that $a_0 = s_0a + t_0b$ and $a_1 = s_1a + t_1b$ at the beginning of each iteration.)

(b) The *inverse* of a mod m, if it exists, is an integer s such that $as \equiv 1 \pmod{m}$. As an example of the usefulness of this algorithm, show that whenever $\gcd(a, m) = 1$, the outputs of Extended_Euclid(a, m) produce an inverse of a mod m. (This is used in the RSA cryptosystem.)

3. A *binary tree* is either empty, or consists of a root node and a “left subtree” and “right subtree”, which are themselves binary trees with no nodes in common. See Figure 8 in Section 9.1 (Section 8.1 in the 4th edition) for an example. Any node in a binary tree both of whose subtrees are empty is called a *leaf.* For example, the tree in Figure 8(a) of Section 8.1 has 6 leaves: f, g, e, j, k, m. The *height* of a binary tree is the distance from the root to the farthest leaf. The tree in Figure 8(a) of Section 8.1 has height 4, m being the farthest leaf from the root. (Note that the distance from the root to m is considered to be 4 rather than 5: it’s the number of edges on the path, rather than the number of nodes.) By induction, prove that for any positive integer n, any binary tree with n leaves has height at least $\log_2 n$. Be careful of the possibility that a node has one empty subtree and one nonempty subtree. (Hint: it will be simplest if your induction mirrors the recursive definition of binary tree given above.)

4. Every day, starting on day 0, one vampire arrives in Seattle from Transylvania and, starting on the day after its arrival, bites one Seattlite every day. People bitten become vampires themselves and live forever. New vampires also bite one person each day starting the next day after they were bitten. Let V_n be the number of vampires in Seattle on day n. So, for example, $V_0 = 1$, $V_1 = 3$ (one that arrived from Transylvania on day 0, one that he bit on day 1, and another one that arrived from Transylvania on day 1), $V_2 = 7$ and so on. Write a recurrence relation for V_n that is valid for any $n \geq 2$. Prove by induction on n that $V_n \leq 3^n$.