Homework Assignment #4 (100 points, weight 5%)
Due: Thursday, April 5, at 1:00pm (in lecture)

Program verification, Recurrence Relations

1. Consider the following program that computes quotients and remainders:

   $r \leftarrow a$;
   $q \leftarrow 0$;
   while $r \geq d$ do
       begin
           $r \leftarrow r - d$;
           $q \leftarrow q + 1$;
   end

   Use the following steps in order to verify that the program is correct with respect to the initial assertion "a and d are positive integers" and final assertion "q and r are integers such that $a = dq + r$ and $0 \leq r < d$".

   (a) Find an appropriate loop invariant that is strong enough to give the final assertion, and prove that it is a loop invariant.

   (b) Using part (a) and other inference rules for program verification, prove the program is partially correct with respect to the initial and final assertions.

   (c) Complete a proof of correctness by formally proving the termination of the loop.

2. (a) Find the characteristic roots of the linear homogeneous recurrence relation $a_n = 2a_{n-1} - 2a_{n-2}$. (Note these are complex numbers)

   (b) Find the solution of the recurrence relation in part (a) with $a_0 = 1$ and $a_1 = 2$.

3. Find all solutions of the recurrence relation $a_n = 7a_{n-1} - 16a_{n-2} + 12a_{n-3} + n4^n$ with $a_0 = -2$, $a_1 = 0$ and $a_2 = 5$.

4. Consider the following recursive procedure to compute the fibonacci numbers:

   procedure FIB(n: non-negative integer)
       if $n = 0$ then return 0
       else if $n = 1$ then return 1
       else return FIB($n - 1$)+FIB($n - 2$)

(a) Set up a recurrence relation that counts the number of times the sum (+) is executed considering all the recursive calls used for input n. (Don't forget to provide initial conditions as well)

(b) Solve the recurrence relation of part (a).

5. Consider the method by Karatsuba for multiplication of large integers given below:

procedure KMULT(A, B, n: A and B are integers with n bits)
1. If $n = 1$ then return $A \cdot B$;
2. else Write $A = A_h 2^{n/2} + A_l$ and $B = B_h 2^{n/2} + B_l$
3. Compute $A' = A_h + A_l$ and $B' = B_h + B_l$
4. $C = \text{KMULT}(A', B', n/2)$
5. $D_h = \text{KMULT}(A_h, B_h, n/2)$
6. $D_l = \text{KMULT}(A_l, B_l, n/2)$
7. return $X = D_h \cdot 2^n + [C - D_h - D_l] \cdot 2^{n/2} + D_l$

(a) Based on the program we can see that the number of basic operations for line 1 is 1 and the total number of basic operations for lines 2, 3 and 7 is at most $C \cdot n$ for some constant C (since the operations are on numbers of at most n bits). Write a recurrence relation for $T(n)$, the number of basic operations used in all recursive calls for the cases in which n is a power of 2 (i.e. $n = 2^k$ for some k).

(b) Use the master theorem (page 479) to find a big-Oh estimate for $T(n)$.