Lab 2 continued

2. Assume the running time T(n) for a particular algorithm satisfies the following recurrence relation:

   T(1) = c
   T(n) = T(n-1) + T(n-1) + T(n-2) + d (for some c, d > 0)

   Use the technique of computing running time for the Fib algorithm discussed in class to solve the recurrence.

3. Below, pseudo-code is given for the recursive factorial algorithm `recursiveFactorial`.

   Algorithm recursiveFactorial(n)
   Input: A non-negative integer n
   Output: n!
   if (n = 0 || n = 1) **then**
   return 1
   return n * recursiveFactorial(n-1)

   Use the Guessing Method to determine the worst-case asymptotic running time of this algorithm. Then verify correctness of your formula.

4. Devise an iterative algorithm for computing the Fibonacci numbers and compute its running time.

5. Find the asymptotic running time using the Master Formula:

   `T(n) = T(n/2) + n; T(1) = 1`

6. *Interview Question.* You are given a length-n array A consisting of 0s and 1s, arranged in sorted order. Give an o(n) algorithm that counts the total number of 0s and 1s in the array. Your algorithm may not make use of auxiliary storage such as arrays or hashtables (more precisely, the only additional space used, beyond the given array, is O(1)). You must give an argument to show that your algorithm runs in o(n) time.