CPS 230 Homework-3

The deadline for handing in solutions is October 16th.

1. **Heap Problems: (20 = 4 + 8 + 8 points)**

   A *d-ary heap* is like a binary heap except that non-leaf nodes have *d* instead of 2 children.

   a. What is the height of a *d*-ary heap of *n* elements in terms of *n* and *d*?

   b. Give an efficient implementation of **EXTRACT-MIN** in a *d*-ary min-heap. Analyze its running time in terms of *d* and *n*.

   c. Give an efficient implementation of **INSERT** in a *d*-ary min-heap. Analyze its running time in terms of *d* and *n*.

2. **Heap Problems 2: (20 = 8 + 12 points)**

   Consider a lazy version of heapsort where each item in the heap is either smaller than or equal to every other item in its subtree, or the item is identified as *uncertified*. To **certify** an item, we first certify its children and then exchange it with the smaller child provided it is smaller than the item itself. (Notice that this is a recursive procedure.) Suppose *A*[1..*n]* is a heap with all items uncertified.

   a. How much time does it take to certify *A*[1]?

   b. Does certifying *A*[1] turn *A* into a proper heap in which every item satisfies the heap property? Justify your answer.

3. **Fibonacci Number Problem:** (20 points)

   Consider Fibonacci numbers defined recursively by *F*$_0$ = 0, *F*$_1$ = 1, and *F*$_n$ = *F*$_{n-1}$ + *F*$_{n-2}$. Prove
   
   $$F_n^2 = F_{n-1} \cdot F_{n+1} + (-1)^{n+1}.$$

4. **Fibonacci Heap Problem:** (20 points)

   Professor Pinocchio claims that the height of an *n*-node Fibonacci heap is *O*(log *n*). Show that the professor is mistaken by exhibiting, for any positive integer *n*, a sequence of Fibonacci-heap operations that creates a Fibonacci heap consisting of just one tree that is a linear chain of *n* nodes.

5. **Recurrence Problems:** (20 = 4 × 5 points)

   Give asymptotic upper bounds for *T*(*n*) for the following four recurrences. Make your bounds as tight as possible.

   a. *T*(*n*) = *T*(√*n*) + 1

   b. *T*(*n*) = 4*T*(*n*/2) + *n*

   c. *T*(*n*) = *T*(α*n*) + *T*((1 − α)*n*) + *cn*, 0 < α < 1 and *c* > 0 are constants.

   d. *T*(*n*) = √*n*T(√*n*) + *n*