Algorithms and Data Structures
Winter Term 2015/16
Exercises for Unit 16 & 17

1. Show the following simple facts about a binomial tree of rank k.
   a) The tree has 2^k nodes.
   b) The root has degree k.
   c) There are $\binom{k}{i}$ nodes at depth i.
   d) If the children of the root are numbered from left to right by $k-1, k-2, \ldots, 0$, then child i is the root of a binomal tree of rank i.

2. In class we discussed the DeleteMin operation for binomial queues. In its realization we processed the rootlist by linking pairs of trees with the same rank until no two such trees remained, and we proved an amortized cost of $O(\log n)$.

   In this problem we consider a different way of processing the rootlist: Consider a maximum set of pairs of equal rank trees in the rootlist, and link each pair. So, a tree that is generated through a link operation in this process will not participate in further linkings (during this DeleteMin operation).
   a) Implement this method in pseudocode.
   b) Prove that this version of processing the rootlist upon a DeleteMin operation also incurs only $O(\log n)$ amortized cost.

3. For the DeleteMin operation for binomial queues we proved $O(\log n)$ worst-case cost, while for Insert we proved $O(1)$ amortized cost.
   a) Construct an example to show that the amortized time for combinations of these operations in binomial queues is lower bounded by $\Omega(\log n)$.
   b) Can we hope for a priority queue that performs arbitrary sequences of Insert and DeleteMin in $O(1)$ amortized time?

4. Define the Fibonacci numbers as $F_0 = 0$, $F_1 = 1$, and for $k > 1$ recursively by $F_k = F_{k-1} + F_{k-2}$.
   a) Show by induction that $F_{k+2} \geq \left(\frac{1+\sqrt{5}}{2}\right)^k$ for all $k \geq 0$.
   b) Show that for all $k \geq 0$:
   $$F_{k+2} = 1 + \sum_{i=0}^{k} F_i$$
   c) Argue that for a sequence satisfying $S_0 = 1$, $S_1 \geq 2$, $S_k \geq 2 + S_0 + S_1 + \ldots + S_{k-2}$, it holds that $S_k \geq F_{k+2}$ for all $k \geq 0$.

5. Professor Pinocchio claims that the height of a Fibonacci heap with n nodes is $O(\log n)$. Show that the professor is mistaken by giving, for any integer n, a sequence of addressable priority queue operations that creates a Fibonacci heap of just one tree that is a linear chain of n nodes.