Data Structures

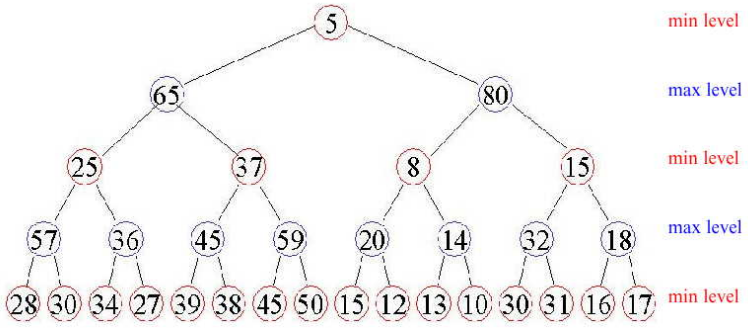
Fall 2019, Homework #4

Date: Dec. 16, 2019

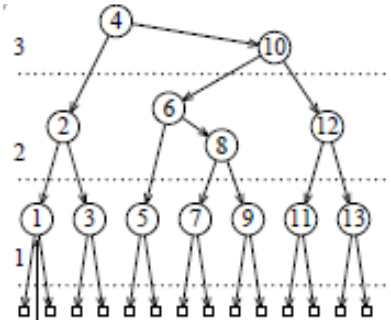
1. (30 pts) The picture below is an example of a min-max heap, a data structure used to implement a double-ended priority queue. A min-max heap is a variation on the normal binary heap data structure. Like a normal binary heap, a min-max heap has the structure property that it must be a complete binary tree. However, the heap order property is a bit different for min-max heaps: every node at an even depth in the tree is smaller than its parent but larger than its grandparent, and every node at an odd depth in the tree is larger than its parent but smaller than its grandparent. A min-max heap supports the insert, deleteMin, and deleteMax operations in $O(\log(N))$ time.

   (a) (10 pts) Where is the node with the minimum priority located in the tree? Where is the node with maximum priority located? Why?

   (b) (20 pts) Give algorithms for the min-max heap deleteMin and deleteMax operations.



2. (25 pts) Consider the following AA-tree. Show how to rebalance the tree after 13 is deleted. Show your work in detail.



3. (15 pts) For any positive integer n, find a sequence of Fibonacci-heap operations that creates a Fibonacci heap consisting of just one tree that is a linear chain of n nodes.

4. (30 pts) We insert the numbers $1, 2, 3, 4, ..., 2^k - 1$ in a Leftist Heap. Let's make that a min heap.

   (a) Suppose that we insert the numbers in increasing order. What is the shape of the resulting Leftist Heap? Why?

   (b) Suppose that we insert the numbers in decreasing order (i.e., insert $2^k-1, 2^k-2, 2^k-3, ..., 3, 2, 1$). What is the shape of the resulting Leftist Heap? Why?