CMSC 341 Homework 4 – Version A

Heaps

Name:
Section:
HW #:
Version:
Username:

1. Show the result of inserting 10, 12, 1, 14, 6, 5, 8, 15, 3, 9, 7, 4, 11, 13, and 2, one at a time into an initially empty min binary heap. You must show each step!!

2. Show the following regarding the maximum item in a min-heap with N items. (Note: A min-heap is a heap with smaller values stored closer to the root.) **Do not give an example to prove the statement. Examples are often helpful but not sufficient to prove.**

   a. It must be at one of the leaves.
   b. There are exactly $\lceil N/2 \rceil$ leaves. (Here $\lceil N/2 \rceil$ is the ceiling of N/2.)
   c. Every leaf must be examined to find it.

3. A min-max heap is a data structure that supports both deleteMin and deleteMax in $O(\log N)$ time per operation. The structure is identical to a binary heap, but the heap-order property is that for any node X:

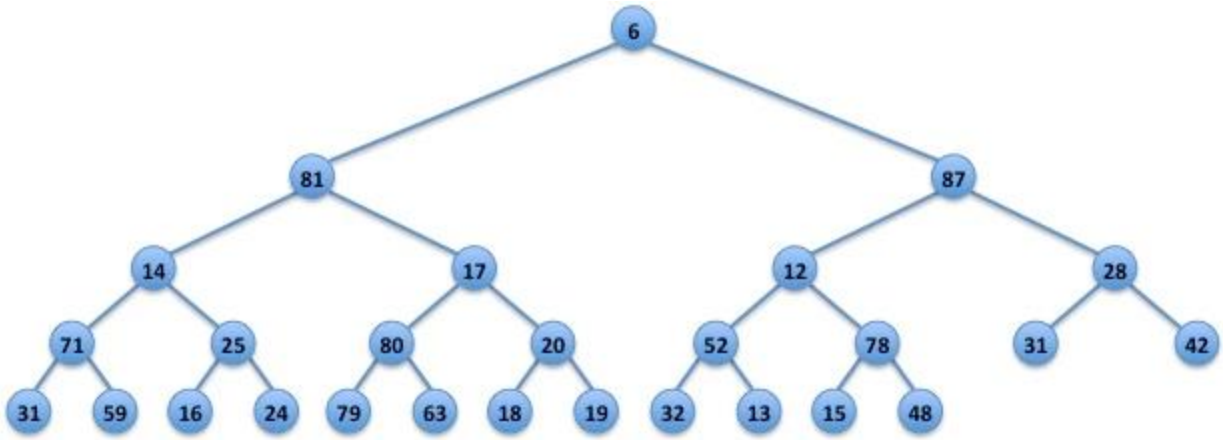
at even depth:

the element stored at X is smaller than or equal to the parent but larger than or equal to the grandparent

at odd depth:

the element stored at X is larger than or equal to the parent but smaller than or equal to the grandparent

See the Figure 1 below.



a. Show what happens to the min-max heap in Figure 1 after inserting 29, 4, 90, 5, 41, 9, 93 and 67. Show your work, then draw a clean copy of the final min-max heap.

   Hint: in a min-max heap, each element is in the interval between its parent and its grandparent. For example, 20 is between 17 and 80; 13 is between 12 and 52; 42 is between 28 and 87. At even and odd levels, the parent and grandparent swap places as the left or right endpoint of the interval.
   Another way to think about min-max heaps is that the even levels form a min-heap where a node's key is smaller than those stored in its grandchildren (up to 4 of them); and the odd levels form a max-heap where a node's key is larger than those stored in its grandchildren. (Note: not all min-heaps and max-heaps interlaced this way make a min-max heap, since interlacing does not guarantee the interval condition.)

   When you insert an item into a min-max heap, you add the new item at the last level, just like in a regular binary heap. However, you have to decide whether the item should percolate up the min-heap through the even levels or percolate up the max-heap through the odd levels. This is determined by comparing the new item with its parent.