Assignment 6 - Solutions

Problem 1

Given a binary tree T containing n keys and a key k, design a recursive pseudo-code algorithm FindSmallerKeys(T, k) that returns a Sequence of keys in T that are less than or equal to key k. For example, given the heap at the bottom slide 34 of the notes and query key k=7, the algorithm should return 7, 5, 4, 6. Note that the keys do not need to be reported in any particular order.

Two possible algorithms should satisfy the above requirement:

1. findSmallerKeys by calling findSmallerHelper
2. findSmallerKeys by calling findSmallerHelper1

Algorithm findSmallerKeys(T, key)

Input: binary tree T
Output: Result Sequence R with elements keys in T that are less than key k R ← new List/Sequence

Return findSmallerHelper (T, key, T.root(), R) // or findSmallerHelper1

Algorithm findSmallerHelper(T, key, p, R)

```
if p.element() <= key then
    R.insertLast(p.element())
if T.leftChild(p)  <= key then
    findSmallerHelper (T, key, T.leftChild(p), R)
if T.rightChild(p)  <= key then
    findSmallerHelper (T, key, T.rightChild(p), R)
return R
```

Algorithm findSmallerHelper1 (T, key, p, R)

```
if T.isExternal(p) then
        return R
if p.element() <= key then
    R.insertLast(p.element())
    findSmallerHelper1 (T, key, T.leftChild(p), R)
    findSmallerHelper1 (T, key, T.rightChild(p), R)
return R
```

Problem-2

Suppose a binary tree T is implemented using an array S, as described in the notes. If n items are stored in S in sorted order, starting with index 1, is the tree T a heap? Justify your answer.

Answer: Yes.

A sorted array will have the lowest value in the leftmost position and largest value in the rightmost position/index.
The heap requirement is:
a) key(v) > key(parent(v))
b) is a complete binary tree externals nodes are filled from left to right.

In an array, the children are always at higher indices that the parent, therefore they will be to the right in the array and have equal greater value than their parents.

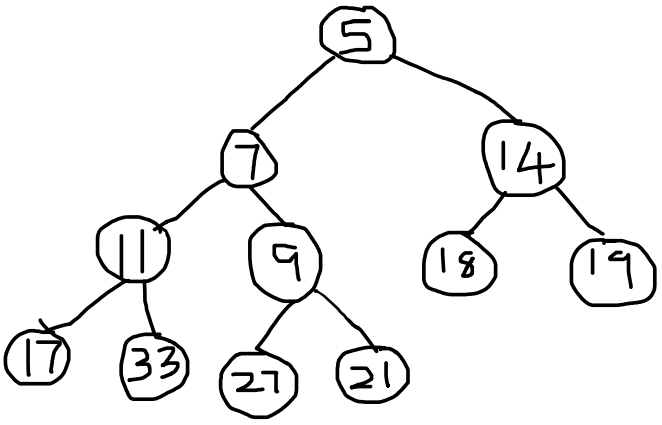
The new elements are added to the right with .push/insertLast and so each internal node is to the left of any external nodes.

My conclusion is a sorted array (lowest to highest is min-heap, highest to lowest is max-heap) is a heap most conveniently/especially if arr[0] is null.

Problem-3

3. Building a Min-heap
a) Insert the following keys into the min heap one by one. You have to capture all the steps with the upheap.

Keys: 9 11 14 5 21 18 19 17 33 27 7



b) Remove the min key from the heap, apply the procedure you learned in the class.