Data Structures **BITL2**

Worksheet 5

14 October 2021

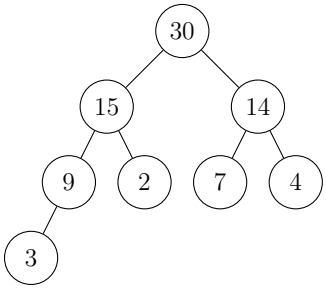
This worksheet uses the following definitions.

- **height of a node:** length of the longest path from a node to a leaf below it
- **height-balanced (AVL) tree**: satisfies $|\text{height(right child)} - \text{height(left child)}| \leqslant 1$

---

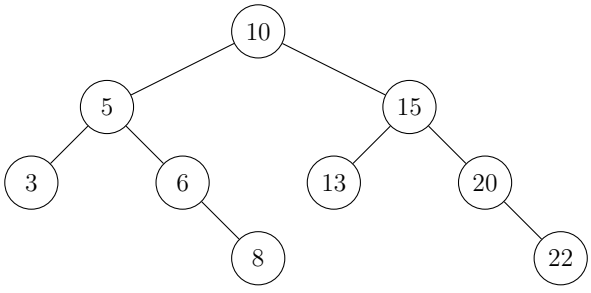
1. **Warm up:** Answer the following questions about tree structures.‘
   (a) What is the shortest and longest path between two level ℓ nodes in a binary tree?
   (b) What is the smallest and largest numbers of leaves a height h binary tree can have?
   (c) What is the heap property of a tree?
   (d) What is the advantage of using heaps for sorting?

2. In the **max-heap** below left, perform the operations below right, in the given order.



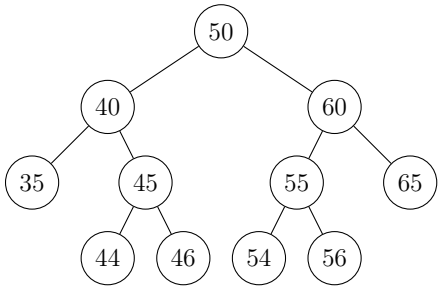
   (a) insert 16
   (b) insert 17
   (c) remove max
   (d) insert 18
   (e) insert 19
   (f) Give the 0-based array that stores the heap after the steps above have been applied.

3. In the **binary search tree** below left, perform the operations below right, in the given order. For each insert operation, show the path taken when searching for the spot to insert the key.



   (a) insert 7
   (b) insert 14
   (c) delete 15
   (d) insert 15
   (e) delete 10
   (f) insert 10
   (g) Give the DFS postorder of the tree after the steps above have been applied.

4. In the **AVL tree** below left, perform the operations below right, in the given order. Make sure to rebalance (if necessary) after every operation.



(a) insert 43

(b) insert 66

(c) insert 67

(d) remove 60

Recall that rebalancing is done in terms of **rotations**. The x**-over-**y rotation of T, for nodes x, y of T where x is a child of y, is a new tree T' identical to T, except for:

- if $T.y.parent = z$, then $T'.x.parent = z$ and $T'.y.parent = x$
- if $T.y.leftchild = x$, then $T'.y.leftchild = T.x.rightchild$
- if $T.y.rightchild = x$, then $T'.y.rightchild = T.x.leftchild$