Data Structures
BITL1

Worksheet 8

Fall 2020

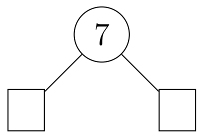
5 November 2020

This worksheet uses the following definitions.

- **binary search tree:** For every node v, $v.key \geqslant v.left.key$ and $v.key \leqslant v.right.key$.
- **height-balanced (AVL) tree**: For every node v, $|\text{height}(v.right) - \text{height}(v.left)| \leqslant 1$

---

1. **Warm up:** Answer the following questions.
   (a) True / False: For every hash function $h\colon \mathbf{N} \to \{1, \dots, 100\}$ there are 100 numbers on which h is constant.
   (b) What is the difference between the *height* of a node and the *level* of a node?
   (c) How many different binary search trees are there for the key collection $\{1, 2, 3\}$?
2. Let T be the following binary search tree (with external nodes marked as squares), and `TreeInsert`(k, x) the algorithm which inserts element x into T at key value k.

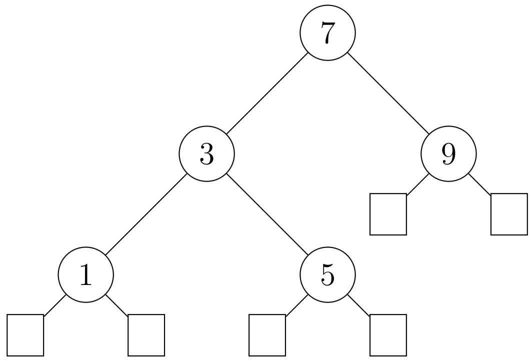


Suppose the commands `TreeInsert`$(k_1, x_1), \dots,$`TreeInsert`(k_6, x_6) are called.
   (a) Give distinct integer keys $k_1, \dots, k_6$ so the commands leave T with height 6.
   (b) Give distinct integer keys $k_1, \dots, k_6$ so the commands leave T with height 2.
   (c) Find a (justified) upper bound on the number of different trees resulting from every sequence of n distinct integer keys.

3. 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 $T'.y.rightchild = T.x.leftchild$

   Let T be the following tree.



   (a) What is the height of each node?

   (b) Suppose that `TreeInsert`$(0, x)$ is called. Draw the resulting tree and the 3-over-7 rotation of this tree.

   (c) Suppose that `TreeInsert`$(4, y)$ is called. Draw the resulting tree T.

      i. Draw the 5-over-3 rotation of T, call it T'.

      ii. Draw the 5-over-7 rotation of T'.