CSC 212: Data Structures and Abstractions
Fall 2018
University of Rhode Island

Weekly Problem Set #9

This is an optional practice assignment!

1 Binary Trees

1. Draw a binary tree after the insertion of the following elements in order:

   `[60, 40, 35, 75, 90, 1, 20, 100, 25, 70]`

2. Draw a binary tree after the insertion of the following elements in order:

   `[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]`

3. Explain what differs in the above two trees. Specifically, address how many leaf nodes there are, the depth of the right, and leftmost nodes, and the height of both the left and right subtrees from the root node.

4. If a Binary Tree is complete, does that necessarily mean it is also full? Justify your answer with drawings of trees.

5. Draw a binary search tree after the following operations steps:

   (a) Insert: [10, 5, 12, 8, 19, 6, 2, 11, 15, 9, 7]

   (b) Remove: [7, 12, 8, 10]

6. Write a function to delete binary trees. Be sure to remove nodes in the proper order, so that none get orphaned.

7. Assume nodes in a BST contain 4 data members: *data, depth, left, right.* Write a recursive function that, given a pointer to the root of a BST, will update every node's *depth* to it's own depth in the tree.

8. Briefly explain the difference between in-order, post-order, and pre-order traversals.

9. Let T be a full k-ary tree, where $k = 2$ (a.k.a. *binary tree*), with n nodes. Let h denote the height of T.

   (a) What is the minimum number of leaves for T? Justify your answer.

   (b) What is the maximum number of leaves for T? Justify your answer.

   (c) What is the minimum number of internal nodes for T? Justify your answer.

   (d) What is the maximum number of internal nodes for T? Justify your answer.

10. Give a $O(n)$ time algorithm for computing the `height` of the tree, where n is the number of nodes.

11. Show that the maximum number of nodes in a binary tree of height h is $2^{h+1} - 1$.