CSC 212: Data Structures and Abstractions
Spring 2018
University of Rhode Island
Weekly Problem Set #11

Due Thursday 4/19 before class. Please turn in neat, and organized, answers hand-written on standard-sized paper **without any fringe**. At the top of each sheet you hand in, please write your name, and ID. The only library you're allowed to use in your answers is `iostream`.

1 Binary Search Trees

1. 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]
2. Write a function to delete binary trees. Be sure to remove nodes in the proper order, so that none get orphaned.
3. Write a recursive function that, given a binary tree, updates every node's value to it's own depth in the tree. You should use two parameters in your call: a pointer to a node, and current depth.
4. Briefly explain the difference between in-order, post-order, and pre-order traversals.
5. Implement a binary search tree with all of the following methods: constructor, destructor, insert, search, remove.
6. 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.
7. Give a $O(n)$ time algorithm for computing the `depth` of all the nodes of a tree, where n is the number of nodes.
8. Show that the maximum number of nodes in a binary tree of height h is $2^{h+1} - 1$.