Sid-Ali Abdullahi

CS 163 HW 4 WRITTEN HOMEWORK QUESTIONS

2/20/2015

**1) Definition of Terms.** In your own words, define the following terms and provide examples of their use:

**a) Full Binary Tree -** a **f**ull binary tree in which all leaves have the same length from the root and are at the same level.

**b) Complete Tree -** A complete binary tree is a bst, which is completely filled, with the possibility of the bottom level, which is filled completely from left to right.

**c) Balanced Tree - a tree is considered balanced** if it meets the following criteria. The tree is balanced if the trees left and right subtrees' heights differ by at most one, AND The left subtree is balanced, and the right subtree is balanced.

2) **Use gdb.** For this assignment, as part of your debugging process use gdb on unix. Write a program through ssh/putty/terminal, compile using g++ on the command line (g++ -g prog1.cpp member\_functions.cpp), and then use gdb to debug (gdb ./a.out). Write a paragraph about how you used it, what it helped you solve problems, and what kinds of features you wish it had. *Hint: gdb is particularly useful for finding seg faults!*

When I used the program to run the gdb at first I didn’t know how to use the program but when I used the program to test it with gdb it gave me a few errors that allowed me to see what was wrong with my program. The program was particularly useful for finding a segmentation that I had in my program.

3) **Algorithm.** Write the complete algorithm for removing an item in a binary search tree. Be detailed. *Use outline form.*

To remove an item from there are three cases to follow.

Deleting a node with no children **:** Simply remove it from the tree.

Deleting a node with one child**:** Remove the node and replace it with its child.

Deleting a node with two children: Call the function until the node to be deleted *is found*. Do not delete *node*. Instead, choose either its [in-order](http://en.wikipedia.org/wiki/Tree_traversal) successor node or its in-order predecessor node, *R*. Copy the value of *R* to *N*, then recursively call delete on *R* until reaching one of the first two cases.