CS 251, Fall 2007
Voicu Popescu
Assignment 8
Due Monday October 29, 11:59PM

AVL Tree

1. Implement a C++ AVL Tree class **AVLTree** in files AVLTree.cpp and AVLTree.h
    a. The items stored are positive integers which serve both as keys and as elements.
    b. The constructor **AVLTree** () builds the AVLTree.
    c. The void **insert**(int *k*) method inserts the key *k* into the AVLTree, and does not do anything if the key is already stored.
    d. The void **remove**(int *k*) method removes key *k* from the AVLTree and does not do anything if AVLTree does not contain key *k*.
    e. The void **printInorder** () method prints the tree to the standard output using an inorder traversal; it prints a (key, height) pair for each node and ends with a newline. Please check the sampleoutput file for format details.
    f. The void **printPostorder** () method prints the tree to the standard output using postorder traversal; it prints a (key, height) pair for each node and ends with a newline; the printing format is the same as for printInorder method.
    g. The destructor **~ AVLTree ()** deallocates all dynamically allocated memory.
    h. Additional instructions
        i. The main.cpp file is provided, but your AVLTree class should work with any valid main file. DO NOT modify the main file, the grader will use both the provided main file and additional main files to test your class, therefore implementing AVLTree functionality in the main file instead of the class files will most likely lead to a 0 for the overall assignment grade.
        ii. Example input and output are provided in the files “sampleinput” and “sampleoutput”. The provided main file takes keyboard input, and you could use “main < sampleinput” on borg machine to redirect the sampleinput file as the input. The output should be printed on the screen and you could also use “main <sampleinput > sampleoutput” to save the output into a file. Please make sure that your program can EXACTLY reproduce that output file given the input file. You may want to use the `diff' program to test this.
        iii. DO NOT change any of the function signatures. Doing so will most likely lead to a compile error when using and a 0 for your overall program grade.
        iv. Be sure to test your program thoroughly.
        v. Please use the Makefile provided.
        vi. Do not partition your code into files we haven't asked for.
        vii. You don’t have to implement **draw**() for this part. Check extra credit instruction for details.

2. Extra-credit [3%] Implement the **draw**(char* filename) function (the function declaration is already in the code provided) that visualizes the AVL tree in a text file named “filename” using the class implemented in lab1. Pixels are modeled with characters. An internal node should be drawn with a 5x3 pixel rectangle, with the key value at its center (the maximum value for the key will be 999 and it will always be positive, so at most 3 pixels will be used for the key value). An external node should be drawn with a 3x3 square. The parent-child link should be drawn with a line segment connecting the centers of the two nodes; however, it should stop at the perimeter of the node. A node should be drawn midway between its two children; ignore half pixel issues. Please check “sampletree.txt” for format reference, this tree has two internal nodes (17 and

148), for extra credit, your output doesn't need to be exactly the same as the sample txt, but it should visualize the tree clearly.

3. Turn in instructions:
    a. Assignment specific:
        i. Turn in files: main.cpp AVLTree.cpp, AVLTree.h(if applicable).
        ii. Your code is REQUIRED to compile with /opt/csw/gcc3/bin/g++ (gcc version 3.4.5) on the borg machines. If your code doesn't compile with this compiler, then even if it compiles with another, your code will still be considered noncompiling.
        iii. Remember that assignment-specific turnin instructions override general instructions.
        iv. Directory structure: all files in the directory lab8.
        v. turnin command executed in the directory containing your lab8 directory:
            - turnin -c cs251 -p lab8 lab8.
    b. General:
        See instruction at http://www.cs.purdue.edu/cgvlab/courses/251/As/turnin.html
        Make sure to check it EVERY TIME before submitting a new assignment. It is updated periodically.