CSCI 241 Fall 2015
Project 4: AVL Trees

DUE 1700 Friday 4 December 2015

AT NO POINT MAY YOU WORK ON THIS PROJECT WITHOUT THE PHYSICAL PRESENCE OF YOUR TEAMMATE(S)

Now that we have explored recursive implementation of insertion and removal from AVL trees in class, it is time to implement them. This final project involves building an AVL Tree that supports inserting and removing elements as well as all three of the traversal methods that we have described. Read the comments carefully in each of the methods. The specifications will be strictly enforced.

In the main section of your Python file (or in a separate testing file if you prefer), provide some test cases to ensure that your AVL Tree implementation functions correctly. Though this is not an exhaustive list, some things to consider are:

1. Does your Tree remain balanced after every operation?
2. Does your Tree remain ordered correctly after every operation?
3. Do your Tree traversal methods return the correct ordering?
4. Do you ignore invalid operations such as attempts to remove items from empty trees?

There will be no extension to this final project. This project will be evaluated on functionality alone (a combination of test results and human-reading of your code). There is no writeup associated with this project. Your submission should consist of a single ZIP file containing a file called AVL_Tree.py and any other scripts necessary to your program or its test cases. The ZIP file should be named after your team, such that a team named Awesome would submit a file called Awesome.zip.