CS 014: Introduction to Data Structures and Algorithms
Fall 2017
Lab 4

Objectives:
In this lab, you need to provide a partial implementation to the binary search tree (BST) and the balanced BST (AVL).

Deliverables:

- (10%) You have to attend the lab on Monday.
- (65%) The final source code.
- (5%) You must adhere to the following submission format. You need to submit a single file on iLearn named "CS014_lab4_<UCR Net IDs>.zip" where <UCR Net IDs> are the students' UCR Net IDs separated by underscore. The ZIP file should contain the source code in .cc, .hh, or whatever extension for source files you used. The ZIP file should also contain a text file named "student_info.txt" that contains the following information:
    - Your TA name.
    - Your lab section number.
    - The name and UCR Net ID for each student in the group.
- (20%) There will be a relevant question during the lab on Monday 10/30 that you will be asked to do during the lab.

Groups:

- This lab should be done in groups of 2-3. Any groups of less than two or more than three students should obtain a permission from the instructor. All group members should be in the same lab section. The group will deliver one ZIP file for the entire group submitted by any group member. The in-lab task will also be done by the same group and only one submission for the group should be submitted.

Due date:

- The deliverables are due on Tuesday 10/31 by 11:59 PM Pacific Time. However, you are highly encouraged to deliver it during the lab on Monday 10/30 to save your time.

Important Note:

- You will need to carry out this lab at home to be able to perform the task that will be given to you during the lab on Monday 10/30. The official due date is on Tuesday just in case you have questions for your TA during the lab.

Problem definition:
BST is a very popular index structure that is designed to support efficient searches and updates. In this lab, you will be given a partial code for BST and your task is to add the missing operations to pass all test cases. The lab consists of two parts. In the first part, you are given a BST with the insert operation implemented and you are asked to implement the erase (delete) operation. In the second part, you are given a partial implementation for an AVL tree and your task is to implement the correct rotation operations to ensure the tree remains balanced after insertion.

Steps:

1. Create a workspace in Cloud9 and initialize it with the code that is attached to this lab. You can do this by pasting the URL 'https://github.com/aseldawy/CS014-Lab4' in the field 'Clone from Git or Mercurial URL.' Instead, you can download the source code from iLearn and upload it to an empty workspace in Cloud9.
   Note: The code might display a warning "warning: overflow in implicit constant conversion."

Please ignore this warning as it is related to the code that prints out the tree and is not related to the tree logic.

2. To compile the code, type "`make`" in the shell. In Cloud9, you can access the shell which is usually at the bottom of the screen.
3. To run the code, type "`./bst`" or "`./avl`" in the shell depending on which part you want to run.
4. The code uses Google Test library to test a few basic cases for tree insert, delete, and balance. Your job is to fill in all the missing parts to make the tests pass.
5. The missing parts are marked with a commented "TODO" item. Follow the hints in these TODOs to complete the required tasks.
6. You should not change the structure of the code. You can create more functions if you would like but you should not remove any of the existing functions.
7. Similarly, you can add more test cases but you should not remove the existing test cases.
8. At delivery, we might use additional test cases other than the ones listed in the source code. Try to make your source code robust by making additional tests.