

CS 2123 Data Structures

Programming Project 5

In this project, we will be implementing the two rotate functions used by AVL trees, and then comparing the running time of an AVL tree vs a regular binary search tree that does not balance itself.

Files provided in the project:

- 1) **BinarySearchTree.h** and **BinarySearchTree.c**. These files contain the prototypes and definitions for all of the functions for a regular binary search tree that does not balance itself. **You do not need to edit either of these files at all.**
- 2) **AVLTree.h** and **AVLTree.c**. These files contain the prototypes of all of the functions we will be using for the AVL Tree implementation. AVLTree has the definitions of all of the functions except for leftRotate() and rightRotate(). **You must provide the implementation of these two rotate functions in AVLTree.c.**
- 3) **abc123Project5.c**. **Rename this file to your abc123.** This file contains a function named compareTreeRunningTimes() which takes as input an integer n. It inserts every number from 1 to n into the regular BST, then searches for every number from 1 to n, and then frees the tree. It then repeats the same process for an AVL tree, and compares the running time. The main() function calls this function with various input sizes. **You do not need to edit this file, although feel free to play with the file to find other ways to compare the trees.**
- 5) Makefile. Update the makefile to reflect your abc123. Compile using **make p5**. Execute the program using **./p5**

Submitting:

Please submit to the blackboard dropbox a zipped folder containing each of the files we have provided you.