Project 2: AVL and BST

CS 3345

By:

Habib Kalia

Janet Eames

Jacob Garcia

Emmett Rasmussen

Introduction

The goal of this project was to first create an AVL tree and organize it by detecting imbalances within the tree after each insertion of “Book” nodes. Each “Book” was set to contain pertinent information in identifying each book allowing it to be sorted among the traditional definition of an AVL tree. The program had to be able to read the given information of each book, and discern the 3 different data points from each other and apply them accordingly. Once each book had been inserted and the program had rotated to balance the tree, the first part of the program was done. From there came the second part, in which it was necessary to create a random binary tree and verify BST order property and AVL balance condition. It was not pertinent to fix any problems occurring, but rather just report them.

Results

The results of the 2 parts were presented by outputting to the console. The first part worked to the fullest extent, as the program managed to go through and balance the AVL tree at every point of imbalance, stating accurately the points in which there was rotation and what rotation operation was performed. The program would read in the information for each book object from the listISBN.txt file, and then create the object and insert a node into the tree. Once all of this was completed, the program would output the height of the tree, and the complexity. From this was presented additional information in regards to position of each node as the tree was traversed with each node showing its left and right nodes, if they exist.

The second portion of the program utilized the list of books as a base and created a random tree of size 100. It utilized a for loop to iterate to 100 while placing each node at the moment it was received rather than sorting it. Once completed, the random tree was judged if it met BST or AVL standards, to which it usually would not, and the program would end.

The text file avl.txt contains a log of the programs output for the purposes of visualization.