Algorithms 4133

HW3

Samuel Lin 010880917

In this Homework Assignment, we were given a mostly-implemented program which tested the functionality of our to-be-implemented binary search tree and AVL tree data structures.

The Binary Search Tree is a non-linear data structure designed to efficiently find a given value. If the given key is greater than the current node’s key, then the search will proceed to the right subtree. This effectively makes finding a given value on average only take O(log n).

The AVL tree is a self-balancing binary search tree. Because of Binary Search tree’s worst case scenario is a time complexity of O(n), the AVL tree was designed to keep the tree balanced (have it self balance itself) upon insertion and removal of nodes to the tree. This essentially means that for any subtree within the tree, there will never be an instance where the difference (balance factor) in height between any left and right subtree is greater than 1. We only had to implement the simple left rotation and right rotations.

These are the results after running the main function through terminal by executing the make file.

This program was implemented and tested on an m1 MacBook Air.

BFS

Text

Description automatically generated

DFS

Text

Description automatically generated

RDFS

Text

Description automatically generated