Jackson Gray / gray344
ECE 368 HW5

Algorithm Description

My program follows the AVL algorithm we've learned in class. Based on the inputted binary streams, each insert recurses down the tree, sends duplicates to the node's left, and updates balance factor when it comes back up. If the balance falls outside of [-1, 1] it performs appropriate rotation to restore a valid AVL structure. After everything is applied and the AVL is checked to be valid it writes the tree in pre-order traversal. It then evaluates by reversing the process and checks again in in-order traversal.

Design decisions/trade-offs

As with previous assignments, I prefer to keep functions consolidated for things like this vs. modularizing. That's just a personal preference and it fits my workflow better. Additionally, on the technical side, though we've talked about how tail recursion is technically more efficient, we are not processing enormous files. Regular recursion is much more intuitive, so I went with that.

Extra Notes

This was hard.