
Professional Integrity Report (CPS101)

Student Name: Alex McColm

Assignment #: 7

It took me 3 hours to complete the assignment.

These parts of the program work well:

The AVL tree is successfully converted to generic. The driver does all specified test operations, on both an AVL tree and a non-balanced binary search tree. The driver successfully copies the dictionary file into both an AVL tree and a non-balanced BST.

These parts of the program don't work well (please identify the specific problem):

Code could be more efficient in the AVL Print class, and likely cleaned up in other places too. I did not attempt the optional part of the assignment this time. There is some duplicate code which could have been averted by using an interface and some more generics.

I learnt the following in doing the assignment:

I saw for myself the sheer efficiency of the AVL tree. When I first loaded the 350ish dictionary words into it, and saw it had 8 levels, I actually thought something must have gone wrong, and there was no way it was *that* efficient. Spoiler, it was.

The difficulties I encountered were:

I copied over the AVL height method by hand, in order to look at it more closely and understand how it worked. In the process, I missed the "+ 1" in a line of the method returning a recursive call to itself plus one. So all my nodes were registering as height 0, and no rebalances were taking place. This was difficult to find, but easy to fix. I had some issues with the Remove method as well, specifically, instead of a node being removed, some other node's value was being copied to root. I didn't understand where that one was coming from, but I looked over the method and fixed it pretty quickly too.

Here are some other comments or suggestions:

I will share my observations regarding the dictionary test. I ended up with an AVL tree of height 8 and a standard BST of height 335. Since the data was inserted in sorted order, the BST degenerated to become essentially a linked list. However, the AVL tree with its rebalance operations, balance factor, and nodes' height members, was remarkably efficient.