Challenge Activity 6

The goal of this challenge activity is to design and execute an experiment which demonstrates the differences in time complexity for several data structures.

# Data Structures

I chose to compare the Binary Search Tree, AVL Tree, and 2-3 Tree data structures.

# Operations

The operations being tested are the insert and search operations.

# Experiment

I generated 50,000, 100,000, 200,000, and 400,000 word lists by concatenating seven randomly selected letters with a normal distribution, then pushing each word onto an array. Each of these lists were copied into a second list. I sorted the second list so that we have a sorted list and unsorted list. I then used the ctime clock function to determine how long each data structure took to first insert each word and find each word for both the sorted and unsorted list.

# Results

It seems that the AVL data structure was by far the fastest at each task. The 2-3 tree was the slowest. I am including my data and charts for ease of further analysis.

# Discussion

There are many opportunities to make this experiment more scientific. The ctime library uses system dependent clock timing, and so my experiment will produce results confounded with system implementation. Furthermore, each data structure may not be a so-called canonical data structure implementation in C++, if such an implementation exists, that structure should be used instead of my own implementation. In retrospect, searching for words in the same order they were inserted in might have skewed my results, further experiments should test search on same and different ordered lists as insert. Testing these structures using strings of seven characters allowed for over eight billion possible words, and with a normal distribution on letters, there were likely few collisions (words being produced twice), however I never conducted any analysis on my domain data.