

This project is broken down into 3 different Jupyter Notebooks--Red Black Final, Scapegoat Final, and AVL Final. Each of these files contains a Binary Search Tree which we tested for insertion time, search time and number of rotations. In order to run the test, you must first run the cells for each different tree. There are 4 different cells to run for the data collection. The first cell contains the methods `timeTrial` and `searchTrial` which find the time for insertion and time for search. The next cell is the “main” code of our test cases; it calls the other methods and collects some of the data in a list for other cells to then create a graph. The final two cells are search time and number of operations. You can change the number of trials by editing the “listOfNodes” list, or by editing the second loop in the second testing cell (the one that creates the insert graph). “listOfNodes” controls how many different n sized trees to make. The second loop--the one with the variable “numberTrails”--controls how many trials are run on a specific tree. To run the visualization code, call the “`my_tree_to_graphviz`” method with a list specified as a parameter, copy the output into a .gz file by entering `vim 'filename'.gv` and pasting the output in. Once the code is in the .gz file type save the changes and enter “`dot -Tpng g.gv -o 'pngname'.png`” into command line.