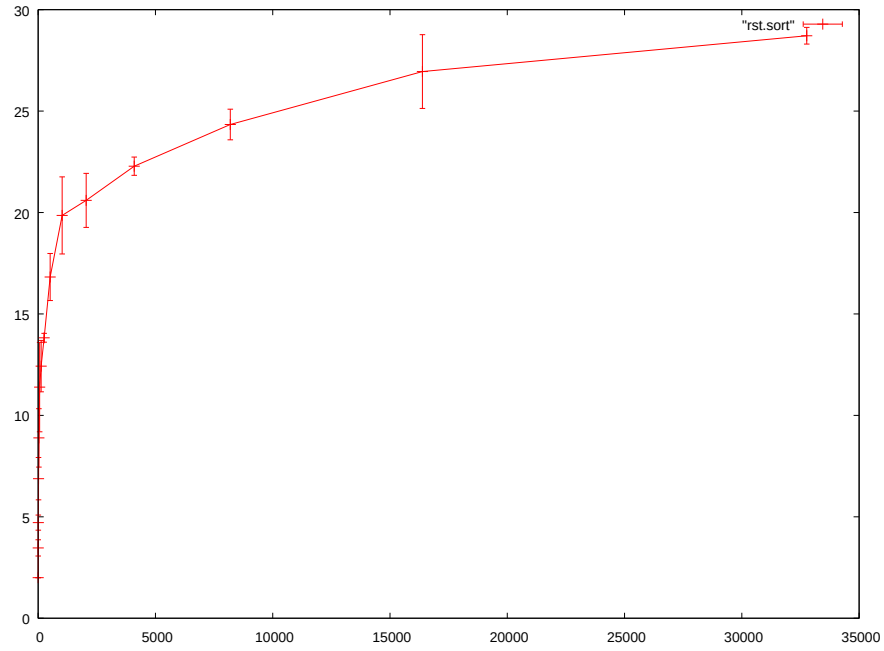
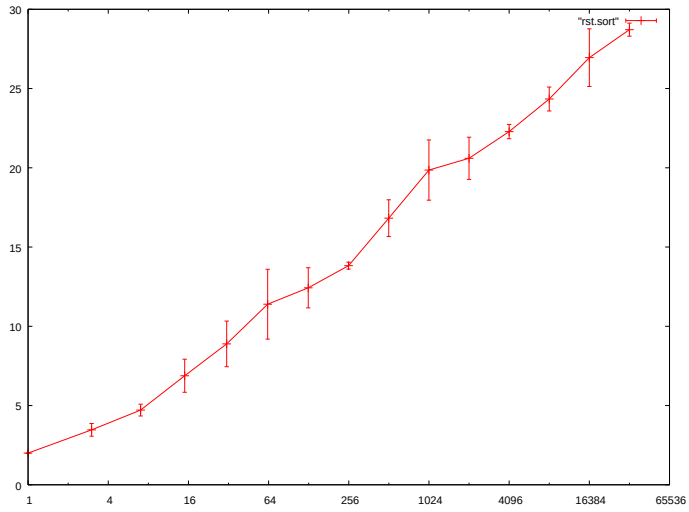
Li, Jimmy
A96026254
CSE 100
Assignment 2

Sorted RST Plot:



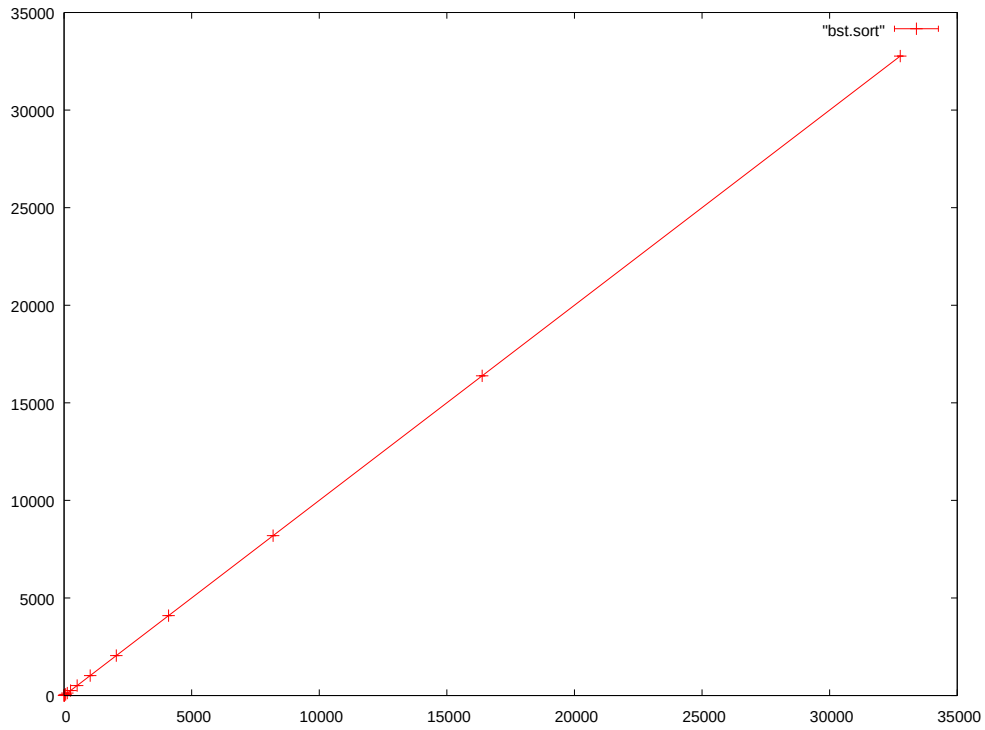
The graph presented shows the average number of comparison (y axis) vs the number of nodes in the data structure. The data agrees with the theoretical big-O time cost of O(logn), as shown in the shape of the graph above. Setting the x axis to a log scale generates the graph shown below.



Because the end result is a linear graph, this further validates the claim that the data agrees with the theoretical time cost. The theoretical time cost of O(logn) makes sense, as RSTs are balanced therefore taking an average of O(logn) to search.

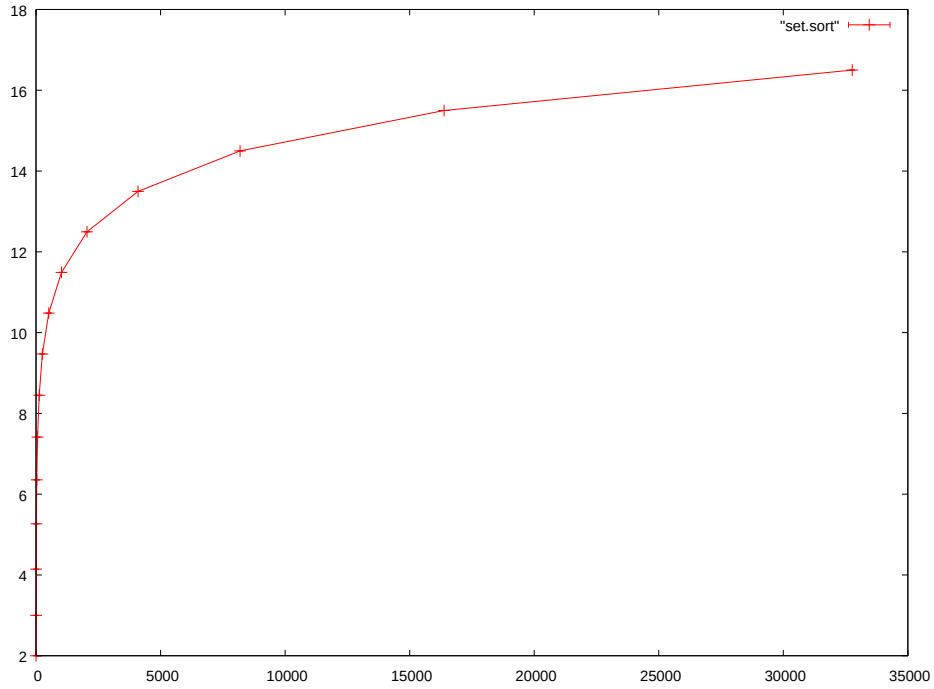
The performance of the RST fares similarly to that of the red black tree, as shown in the sorted set graph. Both exhibit O(logn) behavior with the RST requiring a little bit more comparisons.

Sorted BST:



A sorted BST on the other hand yields a linear time cost of O(n), which agrees with the theoretical time cost as well. BSTs are not balanced and can therefore have its search time lengthened. In this scenario, data is added incrementally, which will skew the BST to the right, making it behave much like a linked list. Because of this linear structure, every node will have to be checked for the find function, thereby giving a time cost of O(n).

Sorted set (red black tree)



The sorted set behaves as expected with a find time of O(logn). It is slightly faster than the RST but as hypothesized they are fairly close.