Name: Scott Chuong and Yukitoshi Murase

Username: schuong and ymurase

File: benchtree.pdf

Description: Benchmarking average number of comparisons for successful find

SORTED RST, Set, and BST:

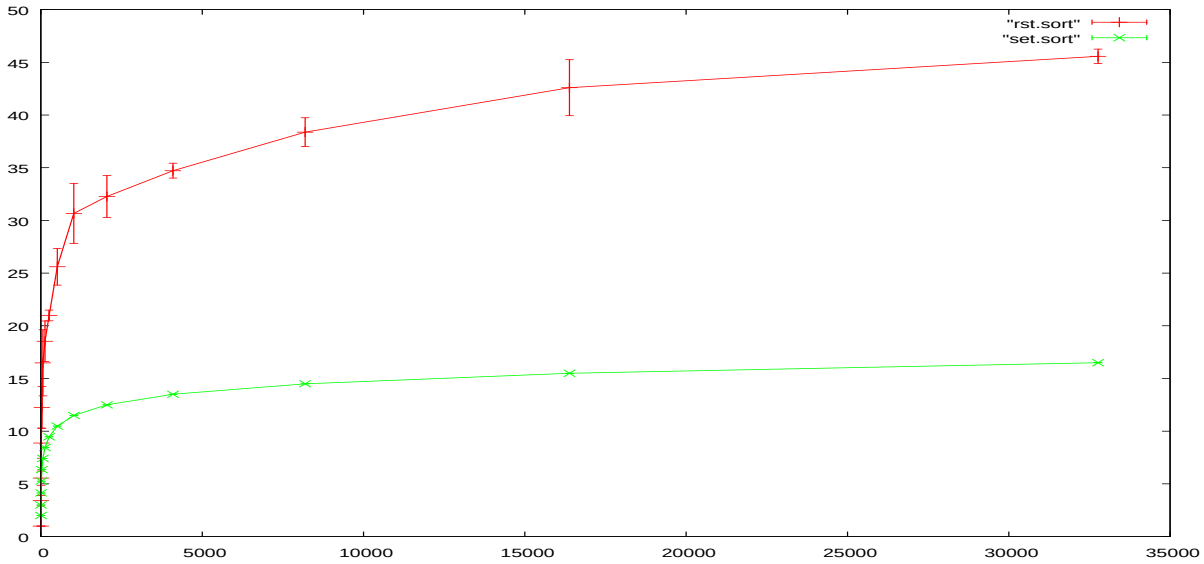

Figure 1: RST and Set Sorted

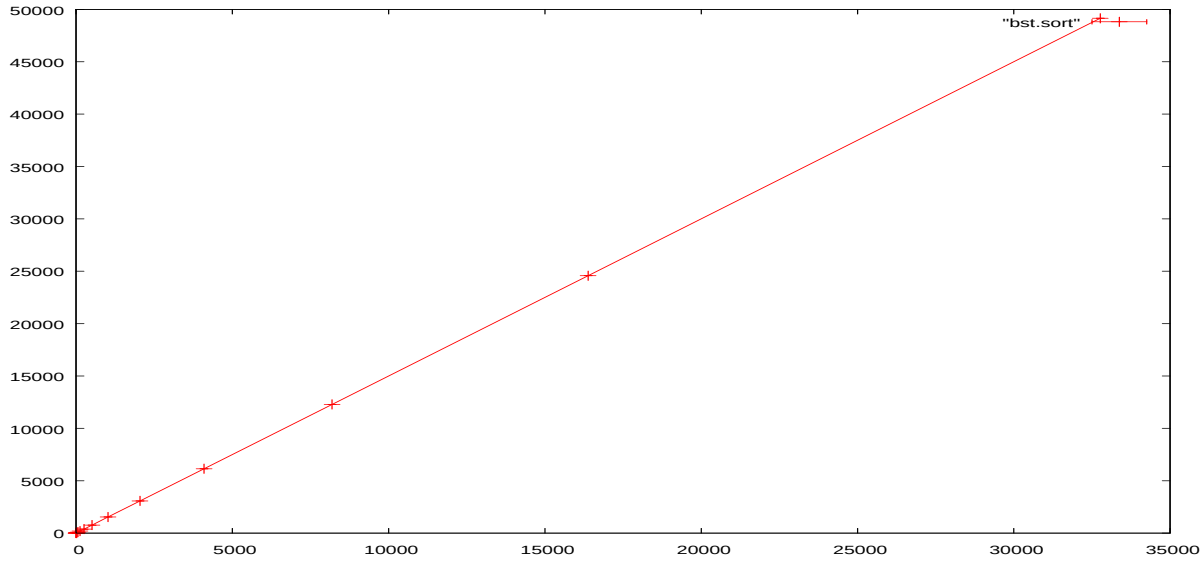

Figure 2: BST Sorted

According to the Figure 1, RST and Set sorted show the time cost of big-oh to be log n. However, in Figure 2, BST sorted show the time cost of big-oh to be n. We believed our program benchmarks RST, Set, and BST sorted correctly because the theoretical big-oh of RST and Set sorted is log n and BST sorted is n. The reason BST sorted takes big-oh n is the data get inserted into linear tree. For example, a sequence of 1-100 will be inserted linearly to the right of the tree. For this reason, during find in a BST sorted, every node will be visited to compare with the search key. However, RST sorted finds search key is quicker because RST insert the data with a randomized key and sort the tree based on the randomized key. Therefore, the height of a RST sorted tree is less than or equal to node -1, whereas the height of the BST sorted tree is equal to n - 1. Comparing to Set sorted, our RST program is not efficient of finding the search key, which can be due to too many comparisons in our algorithm.

SHUFFLED RST, Set, and BST:

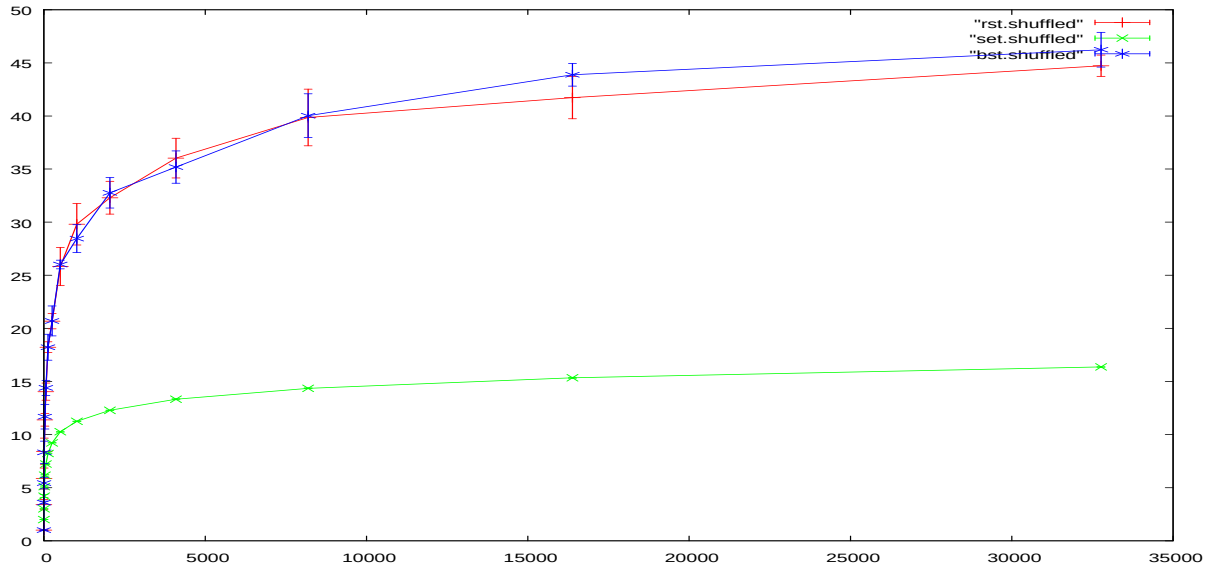

Figure 3: RST, Set and BST Shuffled

Based on the Shuffled graph from Figure 3, all three binary trees have the big-oh of log n. With that said, our program results agree with the theoretical big-oh of log n. Furthermore, BST shuffled is better than BST sorted because the data is inserted in a randomized order. For that reason, there's a high possibility that the BST is not a linear tree. Therefore, when finding the search key in a BST shuffled, not every node need to be visited. Also, BST shuffled are similar to RST sorted and shuffled because its data is randomized. RST sorted and shuffled are similar because of the randomized key used in insert, which prevent the tree from being a linear tree, so it doesn't matter if the data is inserted in a sequence or randomized. Again, comparing to Set sorted, our BST and RST programs are not efficient as set when finding the search key, which can be due to too many comparisons in our algorithm.