

Domenic Mancuso
Assignment 6

Time to build the tree map: 7 seconds

Methods involved:

dateConvert: $O(1)$

Converts string Date into LocalDate

CreateReport: $O(1)$

Converts string from csv into a report object

MapAddMethod: $O(\log(N))$

Using several if/else statements, creates entries in map, innermap, or arrayList depending on if-conditions

Order is $\log(N)$ because of multiple containsKey() implementations, which are from the TreeMap class with complexity of $\log(N)$

ReadCSVFile: $O(N \cdot \log(N))$

Creates N report objects using while loop ($O(L)$)

Adds to treeMap depending on conditions using mapAddMethod ($O(\log(N))$)

Time to process/get selected treeMaps: 0 seconds

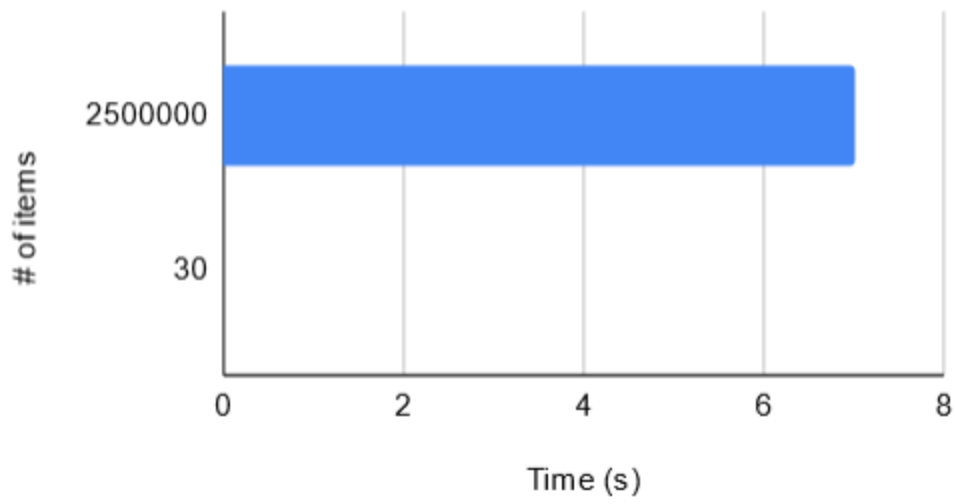
printTree: $O(N)$

Uses for loop to navigate N entries in tree

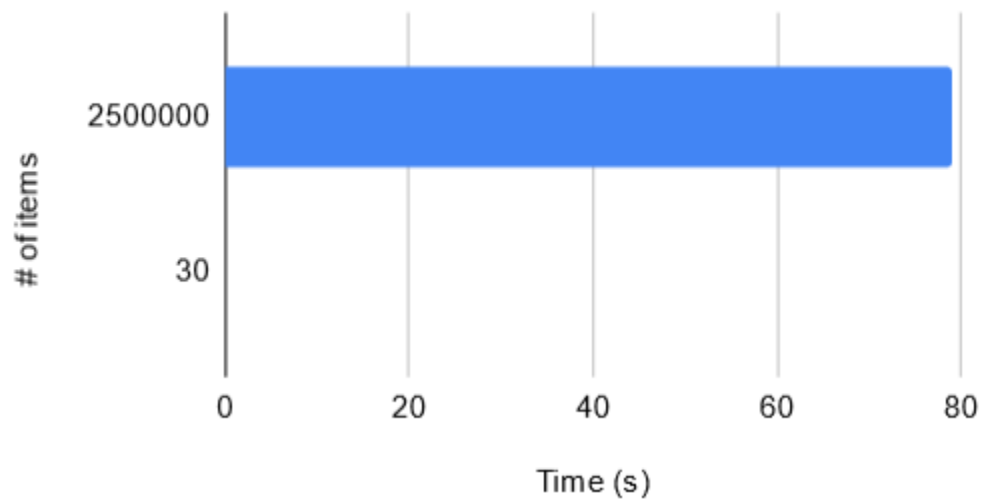
If values in specified state key meet if conditions,
add values arrayList size to count variable

Then, print count

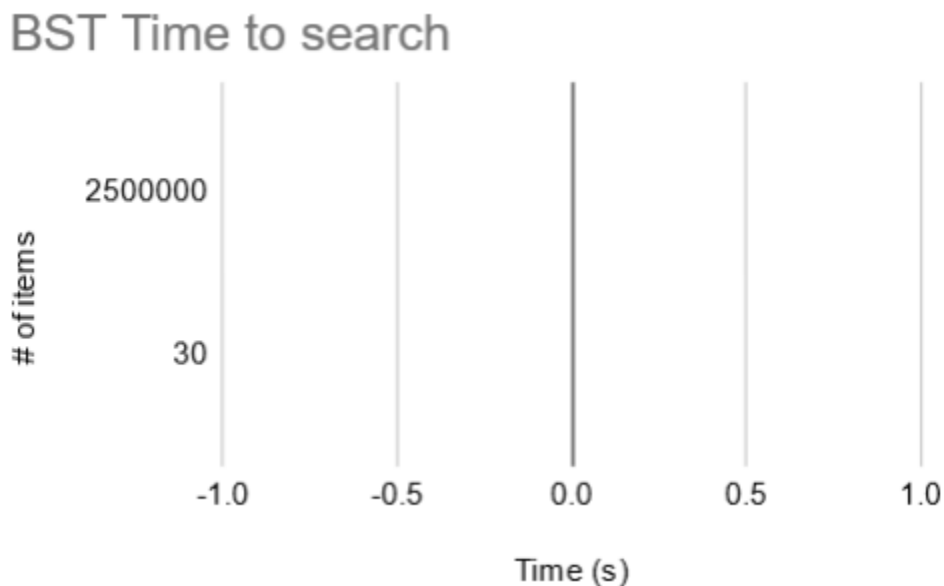
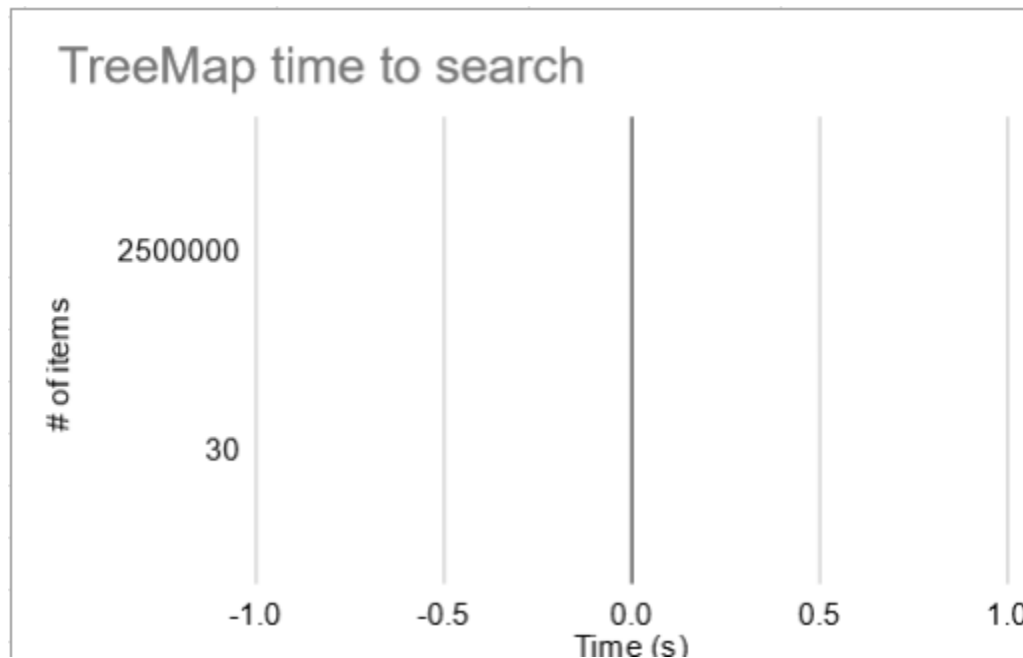
TreeMap time for insertion



BST Time for insertion



As you can see, there is a large difference in insertion time for TreeMaps compared to Binary Search Trees. Seems like Java's implementation of the TreeMap is more efficient.



The difference in search times for both data structures were negligible - both data structure's search methods took less than a second to complete.

Overall, TreeMap is a data structure that was implemented by another person. In theory, you could make it yourself. I believe that the TreeMap was more efficient because it incorporated better code than what was used in the Binary Search Tree, which was made by me within a week. Regarding search, both methods were straightforward in approach, left child having a

lesser value and right child having a greater value is an extremely efficient way to create a sorted data structure.