Ryan Ellis
COSC 220-003
Homework #9: Trees
Section 2.1 – Sorting Analysis
12/5/2024

Recorded Output Findings

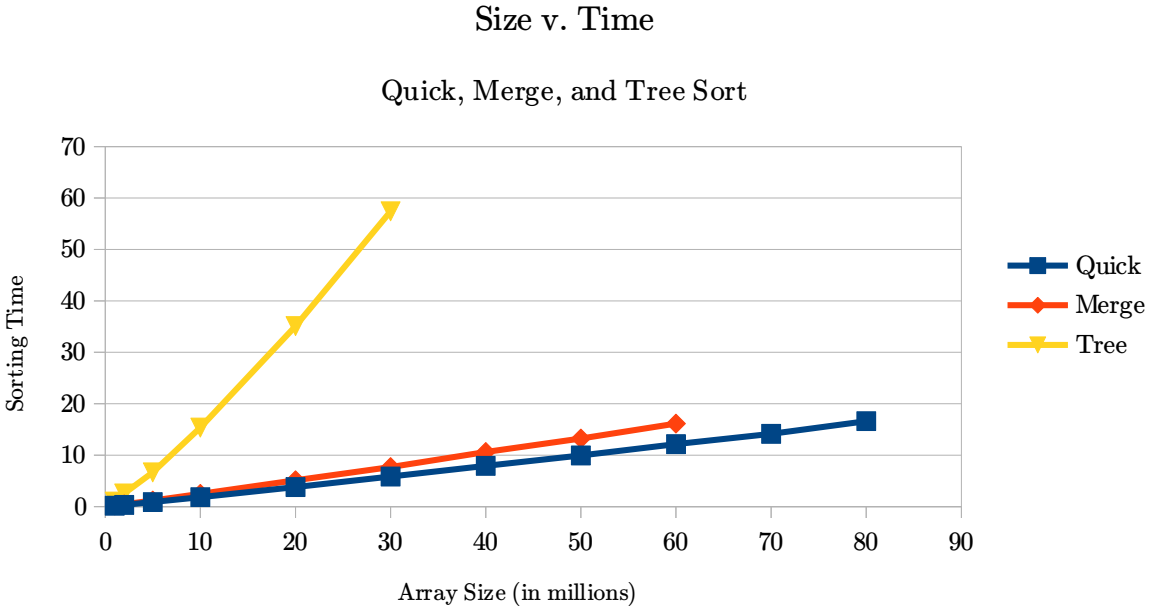

Figure 1: Sorting Time for Quick, Merge and Tree Sort

Analysis

After reviewing the data, it can be noted that the **Quick Sort** has the most consistent efficient performance in comparison to the other sorting methods, resembling a linear form of efficacy. Given that **Merge Sort** is slightly less efficient as the array size increases, still with a linear structure but a larger coefficient of depreciative efficiency, it is very clear that **Quick Sort** is the most efficient among the algorithms. It doesn't appear that any lines of each graph intersect at any point to the naked eye, but it is possible this occurs in the smaller array sizes.

Looking at the data for the **Tree Sort** method, the graph almost has a parabolic curve to it, indicative of a more heavily weighted computational cost for the CPU. Both the **Merge Sort** and **Quick Sort** have similar slopes and linear tendencies, whereas the **Tree Sort** has an almost exponential growth in the amount of time it takes to sort the array once the array sizes begin to drastically increase.