Nader Banna
Professor Rene German
CPSC 350-01
19 December 2020

Assignment 6 Sorting Report

The time differences between the non-recursive sorting algorithms surprised me as the gap drastically changed the larger amount of data I tested the algorithms on. Specifically, with 10 items to sort through, Selection Sort took exactly double the time as Insertion Sort and Bubble Sort, which had identical time of 0.000001s. However, when we move to 1000 items, Bubble Sort has the longest time of 0.002355s while Insertion Sort was the quickest at 0.001598s. The tradeoffs in picking one sorting algorithm over another depend on the specific use case. The factors one should consider are how large is the amount of data that will be sorted as well as the importance of speed vs. simplicity of the algorithms. In the case of Selection Sort, Insertion Sort, & Bubble Sort, they are simple and quick to implement however their runtimes are O(N^2), which is not as efficient compared to Quick Sort which has a best case runtime of O(N log N) and Merge Sort which averages a runtime of O(N log N) as well. For an individual who has larger data sets and a bigger importance for speed, they should choose Quick Sort or Merge Sort for their sorting algorithm. Some shortcomings of this empirical analysis are the lack of direct numbers for comparison on Quick & Merge Sort Algorithms. I was not able to get them functioning properly for the purpose of this assignment. Other possible shortcomings are the lack of comparison of these algorithms between programming languages. I choose to use C++ to complete the sorting however it would be valuable to analyze the specific runtime across various programming languages.