Gurleen Buttar
CPSC350-03

Assignment 6-Sorting Report

The time differences in the sorting algorithms were more drastic than I had expected. Given the options of Merge Sort, Quick Sort, Insertion Sort, Selection Sort, and Bubble Sort, I already had some idea as to which ones would work best with smaller data and larger data. When learning about the different run times of all the sorting algorithms, I wasn't sure how different they would all truly be. In general, Merge sort has an average runtime of O(nlogn), which is the same average time for Quicksort. Both Merge Sort and Quicksort are more efficient than its counterparts, however Quicksort is generally the best sorting algorithm because of its quick runtime and small amounts of constants. Next is Insertion Sort, that does the best job when the data is already partially sorted. Selection sort on the other hand is less efficient and more time consuming and lastly Bubble Sort has the worst time because it only passes through the array once.

The tradeoffs involved in picking the first two algorithms, Quicksort and Merge Sort, are that they are generally more efficient and run times are faster, therefore the best choice for sorting data. Insertion sort's pro is that if the data is partially sorted, it becomes more efficient. Selection sort and Bubble Sort on the other hand are less efficient with no good trade offs when deciding between these two and the other algorithms.

The choice of programming language used was C++ was a great option because it has the ability to run all the different algorithms, without any complex or unnecessary changes. Other languages, such as Java and Python, may also be able to run these algorithms, but there may be differences in efficiency that C++ may or may not account for. The language chosen to run code will always have an affect on the results.

There are some shortcomings with the way this program was created and the tools used. The data sets used were not nearly as big enough as needed to accurately test run times and efficiency. It is best to have large data sets and frequent tests to showcase data accurately. Furthermore, the time algorithms may have some discrepancy in calculating accurate start and end run times.