

Taylor Ropiequet

Rene German

CPSC 350-02

19 December 2020

### Sorting Algorithms

For this assignment I implemented the 5 sorting algorithms: quick sort, merge sort, selection sort, insertion sort, and bubble sort. I used it to sort a text file of doubles. I noticed that when working with a small dataset, around 20 doubles, all the algorithms took under 0.003 seconds. With this small dataset, insertion and bubble sort were the quickest, while merge sort took the longest. When I increased the dataset, insertion sort still remained the fastest while bubble sort's time increased significantly. Insertion sort seems to be the most efficient, because the time hardly changed as I increased or decreased the dataset. I think this is because some of the data may have been partially sorted. It was very noticeable that bubble sort is only efficient for small datasets. Even though bubble sort is easy to implement, I would only use it if I couldn't remember how to implement another sorting algorithm with a small dataset. Selection sort's time also drastically increased as the dataset grew. I think selection sort is most efficient for a smaller dataset. Merge sort worked much relatively quicker for the larger dataset. It was much slower compared to other algorithms for the smaller dataset. Quick sort was the most memory efficient algorithm. I'm not sure if this was the most effective way to gauge the differences between the sorting algorithms because the datasets were relatively small compared to what I could be using these algorithms for in the future. Overall, I think this assignment allowed me to visualize the different sorting algorithms to analyze which would be the most efficient for my needs.