Donavyn Foye

12/12/23

Mini Report

The results of the sorts didn’t shock me in any way besides the Heap sort, Quick Sort, and Merge sort. The sort with the largest time was the Shell sort and the sort with the shortest time was the Quick sort. When starting the project I knew that the Quick sort would be the fastest, however, I didn’t expect the Shell sort to be the slowest. In the graphs, you can see that the number starts off really high and falls off when the other sorts get used. There are some strengths and weaknesses to each algorithm that may have an effect on the speed of the program. The Bubble Sort’s strength is that it is simple and easy to understand and implement. Its elements also are swapped in place without using additional storage. The weakness is that it is not efficient for large databases. It is mostly suitable for academic teaching rather than real-world application. The Insertion Sort’s strength is that it is simple to implement and it is great with small databases. Its weakness is its big O notation and it is only useful when sorting a few items. The Selection Sort’s Strengths are that it is simple and it doesn’t use a whole lot of memory. The weakness of it is that it won’t be efficient for large datasets and it’s a big O notation. Merge Sort’s strength is that it is consistently efficient and can handle large datasets decently well. Its weakness is additional memory requirements. Shell Sort’s strength is that it is adaptable to the different data thrown at it. Its weakness is that its big O notation depends on the chosen gap sequence. The Quick Sort’s strength is that it is able to deal with a huge list of items and because it sorts in place, no additional storage is required. The weakness of Quick Sort is that its worse-case performance is similar to the average performance of bubble sort, insertion sort, and selection sort. The Heap sort’s strength is that its memory storage is minimal and it can be implemented as an in-place sorting algorithm. Heap sort’s weakness is that it requires more space for sorting and Quick sort is usually more efficient than Heap sort in most cases.

In conclusion, this project showed me the efficiency of each of the sorts that were tested in the project. I learned more about their implementation and their big O notation as a whole. It was also interesting to see the collection of data in graph form as well.

A graph with a line

Description automatically generatedA graph with a line

Description automatically generated

<https://www.bigocheatsheet.com/>

<https://dbmspoly.blogspot.com/p/advantage-disadvantages-of-sort.html>