**SD Assignment 1: Intro algorithms & Datasets**



Student: Johnson Domacasse

Date: 15 sept 2023

Student#: 4471709

Teacher: Lake Lakeman

## Introduction:

There a many different types of sorting algorithms. Each with their own pros and cons. In this small document I will explain what I have found while using 3 different sorting algorithms. In the procedure I will talk about which algorithms I chose and for which reasoning.

## Procedure:

### Bubble sort:

I chose bubble sorting as my first algorithm is because it is the algorithm that I have used once or twice before. I know from experience that this algorithm has a major flaw. It can get quite slow while working with bigger files. So what I am expecting is that this algorithm will take longer with larger files but shorter on smaller files. I will discuss what I found in the conclusions section.

### Quick sort:

I chose quick sort because in the previous assignment I used it and noticed a considerable increase in speed compared to my first program. After doing some research, I see that can function with n log n notation on two different occasions but its worse is n squared. So I’m curios to see how this algorithm function with my given files. I will discuss what I found in the conclusions section.

### Merge sort:

I chose merge sort because while doing research, I noticed that this was another algorithm that is one of the if not the best algorithm I found. The only noticeable issue I saw was that compared to the rest, this algorithm has a very poor memory. I will discuss what I found in the conclusions section.

## Conclusion:

The tests were all successful. The only change you can see between them all is the time. This was expected. My results were more or less what I expected.

### Bubble sort results:

The bubble sort was almost faster than both of the other 2 algorithms when it came to the smallest file. But as the files grow bigger, the speed at which it sorts them gets slower. What surprised me was the fact that it ended up being less than a second. I expected just above a second.

### Merge and quick sort results:

As you can see below the quick sort algorithm is slightly faster when it comes to bigger and smaller files. However this is my first test (see below). After running the test 2 more times, quick sort had the best time in all 3 cases.

A screenshot of a computer

Description automatically generated

My conclusion here is that currently I am working with simple datasets provided to me so I will not be able to draw a concrete conclusion on which of my chosen algorithms are best for which case. What I can say is that if you are making a system that prioritizes speed over memory then quick sort would be your ideal option. However right now I am neglecting the worst case scenarios for the algorithms and also the stability of them. More can be read form the link below on sorting algorithms.

## References:

<https://en.wikipedia.org/wiki/Sorting_algorithm>

<https://betterexplained.com/articles/sorting-algorithms/>

<https://www.geeksforgeeks.org/bubble-sort/>

<https://www.geeksforgeeks.org/merge-sort/>

https://www.geeksforgeeks.org/quick-sort/