Algorithm Analysis and Design Project

Imam Abdulrahman Bin Faisal University | College of Science and Humanities | Computer Science | Supervised by : Dr. Azza A. Ali

Introduction

In our report, we tried to describe the implementation and analysis of each type, as well as the best, average, and worst cases. The three alternative methods were analyzed based on their time complexity using inputs of different sizes. In addition, the Java programming language was used to implement algorithms.

Problem

Nowadays, the speed of computer execution of algorithms has become essential for faster program execution. Still, it is possible that some people do not care about the speed of computer execution but rather the memory space occupied by the program. Therefore, choosing the algorithm that is faster to run and performs better is very important. Therefore, our task in this report is to explain the difference between the algorithms in terms of runtime to display the best among them and the most appropriate.

Result

X-axis represents input size, and y-axis represents median time (in microseconds) in Figure 1, 2 and 3.

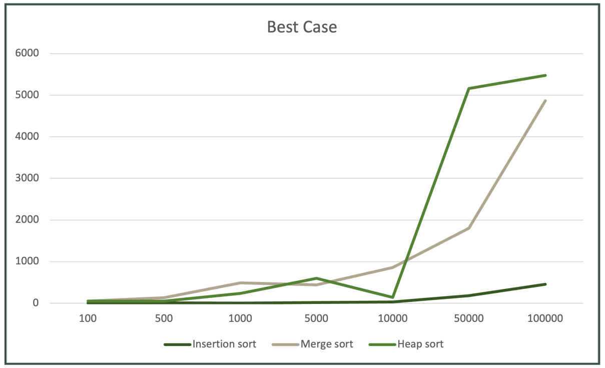

Figure 1: Best case of all sorting algorithm

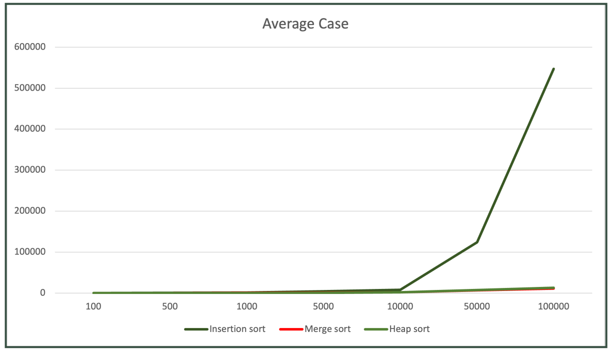

Figure 2: Average case of all sorting algorithm

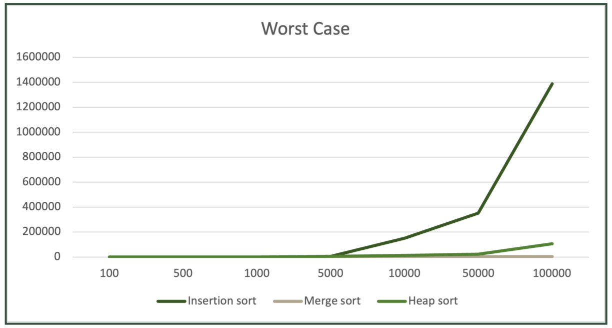

Figure 3: Worst case of all sorting algorithm

Discussion

It is essential to consider large input sizes when analyzing the best running time. Since we always consider the worst case, we conclude that merging sort is the fastest sorting algorithm. Theoretically, heap sort is more efficient than merge sort because it doesn't require massive recursive calls and multiple arrays. On the other hand, merge sort is faster than heap sort if the input size is large. As a result, the merge sort is the most efficient. Despite the merge sort algorithm's efficiency, for small input sizes, the fastest algorithm may vary. Also, if the list were sorted, insertion sort would be the most efficient sorting algorithm.

Conclusion

In this report, we try to summarize nearly all the sorting algorithms. Therefore, to sort a list of elements, First of all, we analyzed the given problem, i.e., the given situation is of which type (small numbers, large values). After that, we apply the sorting algorithms but keep in mind minimum complexity, minimum comparison, and maximum speed.

Group Member

- May M. AlOtaibi
- Ghala M. Alkhaldi
- Fida M. Alelou
- Bushra M, Alshehri
- Jumana Y. Aljassim