Name: Mulia Widjaja (Noble)

Instructor: Dr. Kai Lukoff

51282 Computer Engineering 12L: Abstract Data Types & Structures Lab

12/2/2022

Lab 5 Report: Sorting Algorithm Runtime Analyses

In this lab, I was instructed to analyze a runtimes of quadratic ($\Theta(n^2)$) and linearithmic ($\Theta(n \log n)$). I chose Selection Sort for the Quadratic runtime. On the other hand, I chose Merge Sort for the Linearithmic Runtime. This document represents the 2-page compilations of analysis along with a table of each algorithm with different numbers or rows of data, referred to as n. The variable n ranges from 10 to 1,000,000. Here are the table below:

	$\Theta(n \log n)$ algorithm (Merge Sort)	$\Theta(n^2)$ algorithm (Selection Sort)
n = 10	0.254 sec	0.249 sec
n = 100	0.263 sec	0.256 sec
n = 1,000	0.267 sec	0.269 sec
n = 10,000	0.298 sec	0.348 sec
n = 100,000	0.744 sec	5.597 sec
n = 1,000,000	5.246 sec	8 min 16.374 sec

The code for the Selection Sort with $\Theta(n^2)$ is apparently simpler, especially with the for loop. Nevertheless, based on these observations above, the Merge Sort method for $\Theta(n \log n)$ is far more efficient than $\Theta(n^2)$ with the Selection Sort one.

With $\Theta(n^2)$ algorithm (Selection Sort), while the number of data is 10,000 or less, the data could have been sorted in less than 1 second. Moreover, a data amount of

100,000 demonstrates that the code can be implemented in approximately 5 seconds. By the time it reaches 1,000,000, it takes over 8 minutes for the code to be implemented.

Contrariwise, with Θ(n log n) algorithm (Merge Sort), while the number of data is 100,000 or less, the data could have been sorted in less than 1 second. By the time it reaches 1,000,000, it still takes solely approximately 5 seconds for the code to be executed. It vividly demonstrates that the code for the Merge Sort method with Θ(n log n) algorithm is much more efficient compared to the Selection Sort one.