Name: Alex Henabray
Course: CS 2420
Assignment 05

Analysis

Question 1

My partner is Jana Klopsch. She submitted the source code for our program.

Question 2

I think Jana deserves full credit for this assignment. She contributed many creative ideas, and she found several bugs hidden in our code from the quick sort method. She is a very reliable programming partner, and I would like to work with her again on future homework assignments.

Question 3

Although I prefer solo programming, I enjoyed writing code with a partner because it was a great opportunity to form connections with a student who shares an interest in computer science and mathematics. I think one of the most useful aspects of paired programming is that both partners often have a unique /different approach to solving problems encountered in the homework assignments. This often helped my partner and me identify bugs and write code fairly quickly. One thing I did not enjoy about pair programming is that there were times where I struggled to navigate while my partner was typing because I didn't know how to implement specific portions of the quick sort algorithm. I felt like I was holding my partner back because she grasped the concepts while I was struggling to understand what we were trying to do.

I think I can become a better programming partner by my spending more time studying the course material. I can prepare for paired programming sessions by reviewing the data structures and algorithms we will be implementing in the current homework assignment, and to write pseudocode for the methods we will be writing.

Part 4

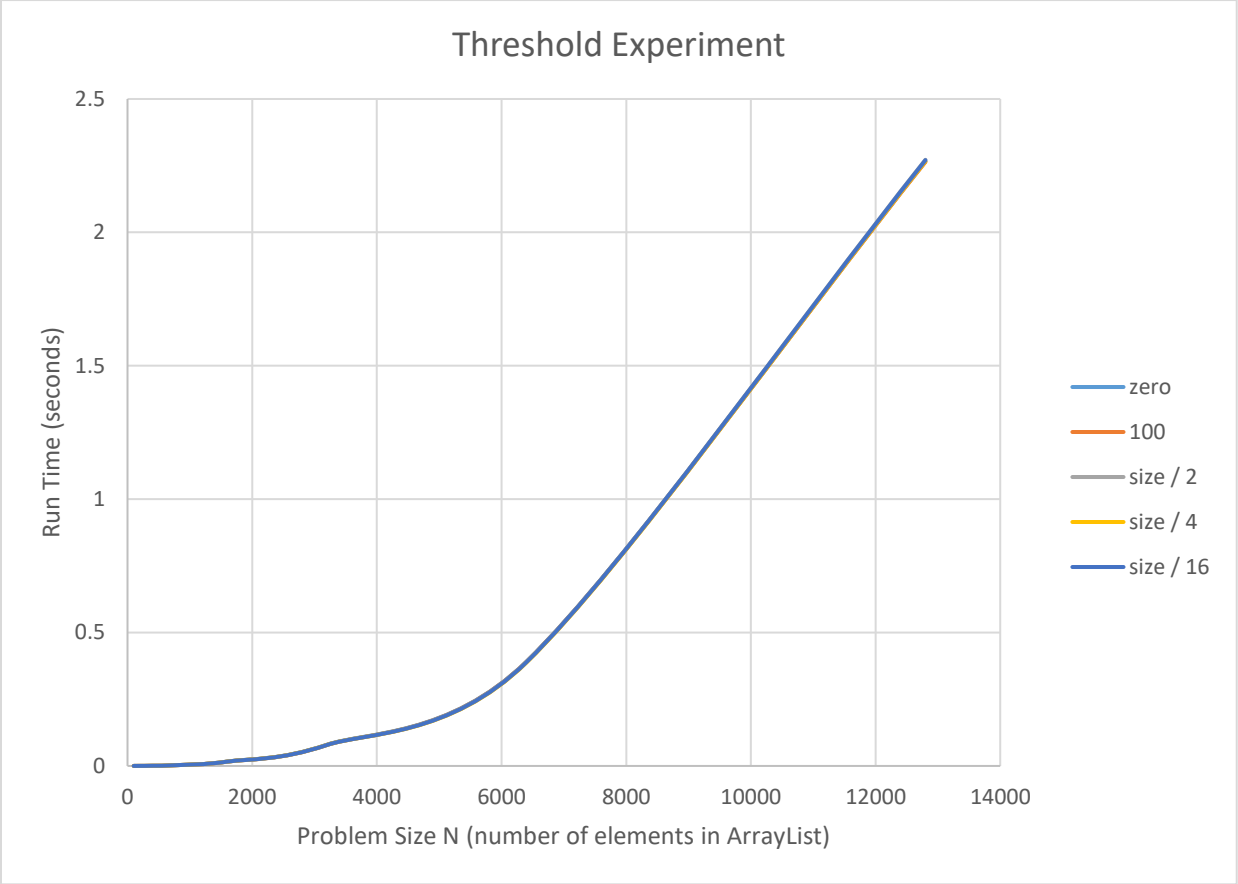

Figure 1

Note: I used the generate best case method to create the array list for the threshold experiment.

Comment: It is difficult to distinguish which threshold value worked best in the merge sort method, but I suspect that I will obtain different result if I used the average / worst case method to generate the array lists for this experiment. Due to time constrains, I was unable to run this experiment again using worst and average case scenarios.

Part 5

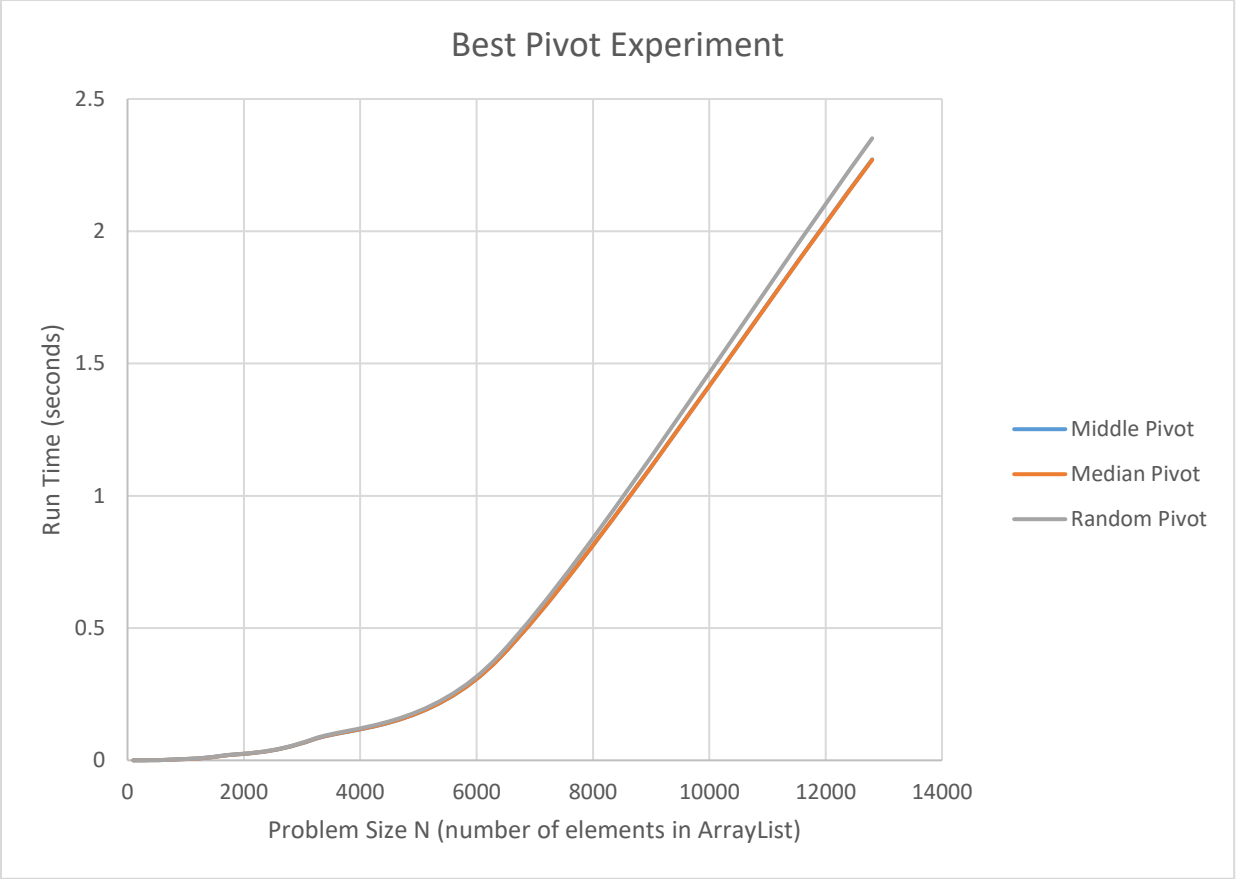

Figure 2

Comment: Middle and median methods worked best for selecting the pivot in the quick sort method.

Part 6

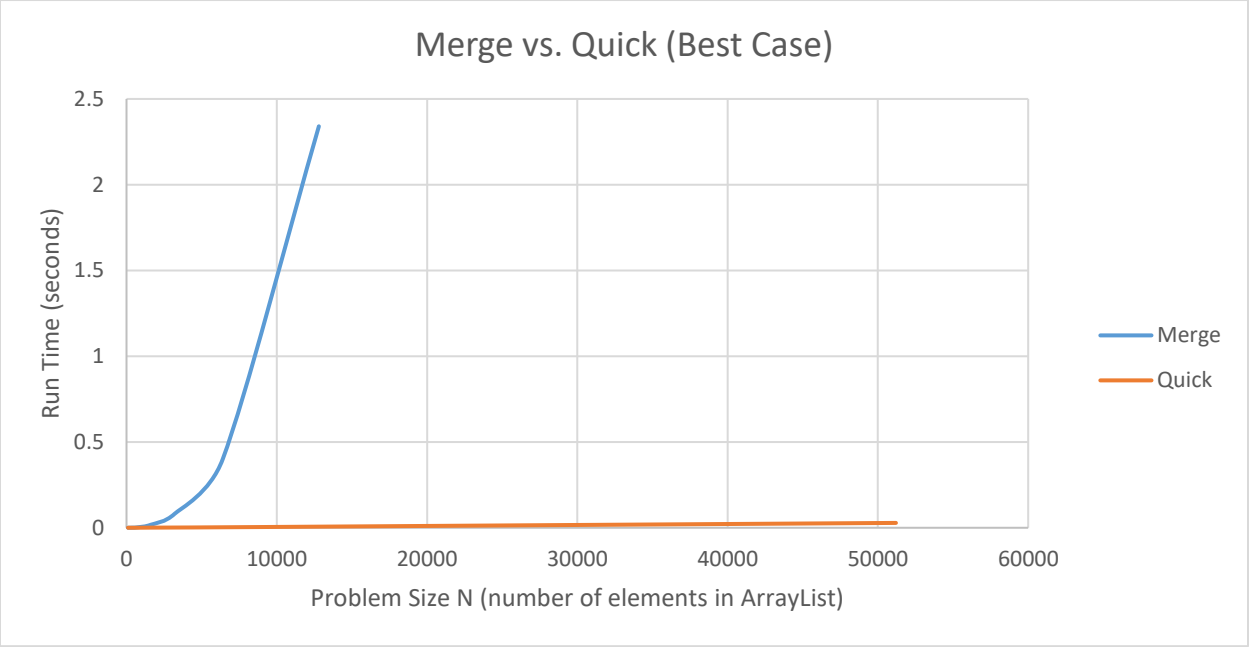

Figure 3

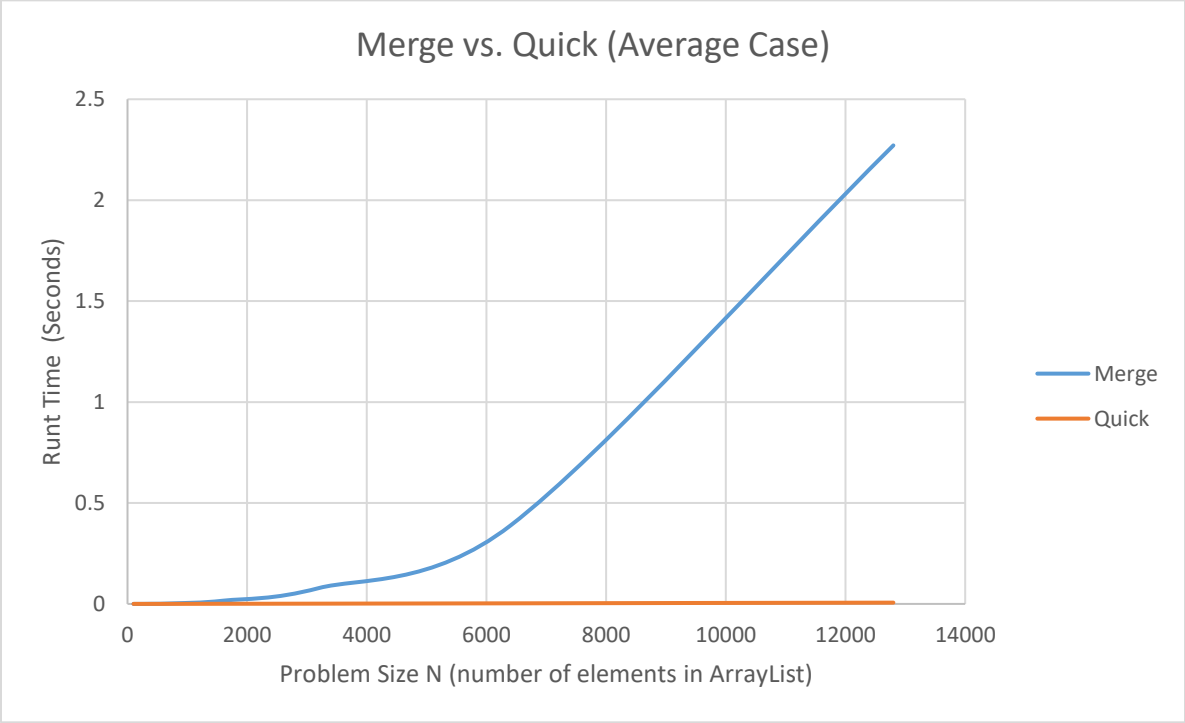

Figure 4

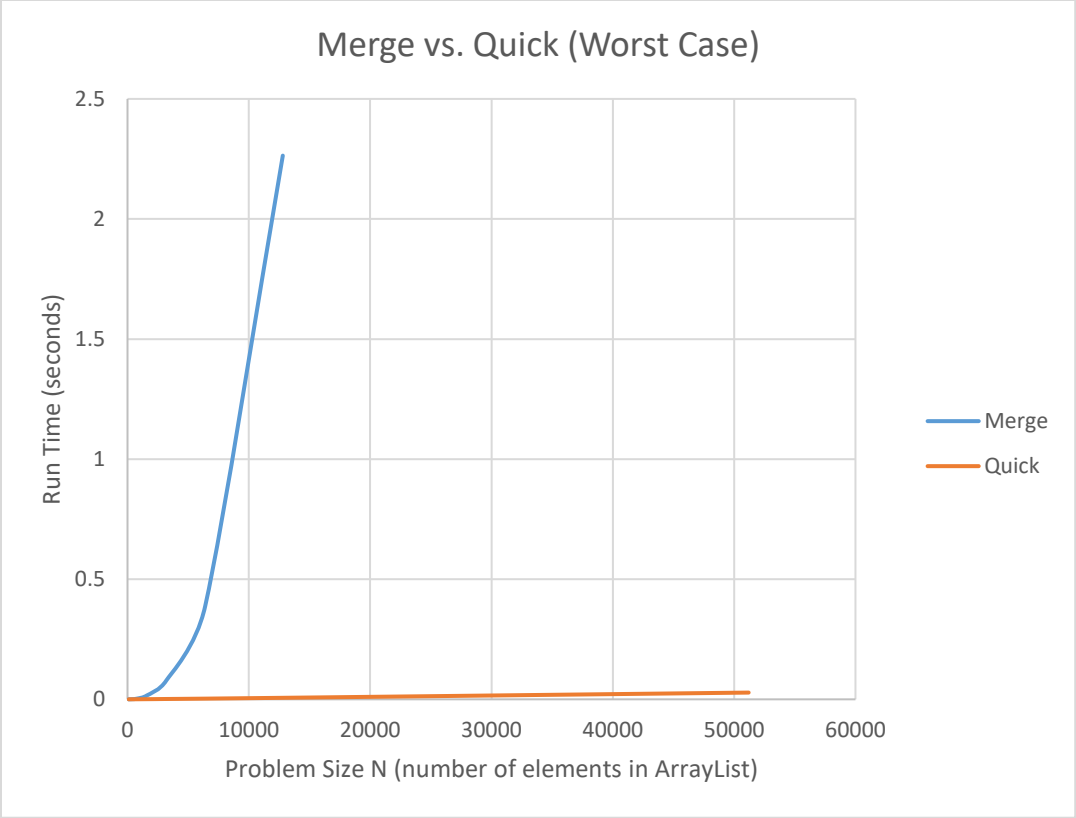

Figure 5

Comment: Based on figures 3 – 5, it appears that the merge sort method has a Big-O behavior of N^2. This makes me suspect that a bug is hiding somewhere in my code.

Question 7

For part 5, it appears that the merge sort method has N-log N behavior for large N, which is what we should expect for the worst, average, and best case. I think this occurs because the method will have to call merge sort recursively approximately N times. Each time the method is called, the input array list is divided in half. By the halving principle, we should expect the complexity of this particular behavior to be log N. As a result, the overall complexity of the merge sort method is N - log N. It does not appear that the different threshold values had a significant impact on the asymptotic behavior of the merge sort method. This seems correct because all of the array lists generated in this test used the generate best case method and the insertion sort has a complexity of N for best cases.

Based on my graphs for part 6, it is hard to determine if the quick sort method has a complexity of N-log N or N^2. One way I could solve this problem is by plotting more points for larger values of N. Due to time constraints, I was not able to do this. It appears that both the middle and median pointers worked best for the quicksort method. This is reasonable because both methods

define in the SortUtil class select the item found in the middle of the array. The only difference between these two methods is that median selects the central element from a sorted array list.

Based on my graphs for part 7, the quick sort method performed a lot better than the merge sort method in all cases. Although the merge sort has passed all of our Junit tests, I do think it is likely that a bug exists in the merge sort method my partner and I wrote because the merge process is showing quadratic Big-O behavior rather than N-Log N behavior.

Question 8

My partner and I spent about 8-10 hours writing the code. I spent an extra 6-8 hours working on the analysis on my own.