Kira Halls

U1250109

9-8-2022

CS 2420

Analysis Document—Assignment 2

1. My partner on this assignment was Braden Campbell. He was the partner that submitted the assignment on gradescope on behalf of both of us. I actually enjoyed both roles of pair programming, but if I had to choose one, I think I enjoyed being the driver a little better. I learned a lot by listening to my partner and how he suggested implementing the code. It gave me a chance to step back from worrying about the overall idea of the code and instead focus on learning from him and making sure the code was written and formatted properly. My partner and I did fairly well with applying the pair programming techniques, with one exception—as this was our first assignment, we struggled to find the right platform that would allow remote access to actually switch between driver and navigator. We switched back and forth a few times, but our second coding session we couldn't get the remote access functioning correctly to actually switch. However, we still tried to switch roles as best we could and supported each other by both brainstorming ideas for the code and correcting each other no matter what role we were in. I feel that my partner and I were very efficient in implementing the program, we got right to work and we communicated very well. It only took us about five hours total to complete the actual code as we were usually on the same page about how to implement it, and we were good to listen to each other and communicate our different perspectives. As mentioned above, it took about five hours to code and test this assignment. Though we didn't keep a specific log of how long we spent coding vs. testing, I'd guess we spent three hours testing and two hours of actual coding. I liked my partner, he was willing to work just as hard as me and was very receptive to my perspective and helped me when I was confused about certain aspects of the code. I think we both communicated well with each other and worked an equal amount on the project. We are planning on continuing working together and we will work on using a different platform so that we can properly switch between driver and navigator. I had absolutely no issues while working with Braden and I think he is a great partner.
2. Comparable is for objects that have an obvious ordering. For example, integers have an expected order, or words have a specific alphabetical ordering. Comparable compares objects that we already know how they are supposed to be sorted. Comparator is used when the objects don't already have a specific ordering. We can implement it to create our own ordering based on how we need the information to be sorted. Comparable is good for instances that have a typical order in the world—dates, words, times, etc. Comparator is good for objects that aren't as obvious—colors, shapes, objects with multiple attributes, etc. We can change the extra features in LibraryGeneric to use Comparable if we use comparable at the core level. Each of the attributes of Library are

things that can have a natural ordering—the ISBN is just numbers, the title is just a string, same thing with the author. Even though they represent a book with multiple attributes—which means it doesn't have a natural ordering—each attribute is a type that can have a natural ordering.

3. 

4. The algorithmic complexity of this lookup method seems to be linear, because we run a for loop i number of times, record the midpoint time, then go through another for loop i number of times. This means if we increase i, we increase the loop iteration i times. We don't multiply the first for loop and the second for loop together, instead we seem to add them which implies linear complexity. Even though the graph doesn't look exactly linear, it doesn't increase fast enough to imply quadratic complexity. It most closely models a linear complexity in both code and plot.