Kara Douville
u0305506

Analysis of Assignment 3

My programming partner for this project was Anthony Sciammarella and he also turned in the source code of our program. We did a good job of making sure we switched ever 10-15 minutes or so and for me this was an appropriate amount of time to spend at each position. We were able to switch so frequently because we were utilizing an Eclipse plugin called Saros. This allowed us to turn our eclipse IDE into something that acted like a Google Doc, where we could both be on different computers but work on the same code. Although we both had the ability to type code at the same time, we made sure that someone was definitely in the driver's and the navigator's position and that we weren't typing code at the same time. This helped a lot to be able to switch more frequently and more effectively as there was no physical moving of chairs or equipment. Also, it was nice that we could employ the "over the shoulder" method, without actually having to be right on top of the other person to clearly see the code. Good communication between us made this method possible. We had to clearly state who was going to type, as there isn't the actually physical barrier of just one keyboard to make sure only one person is typing. Also, we both have very different schedules, and without this technology we might not have been able to make this pair work. Fortunately because we can work form different places and then communicated either via phone or Skype, we were able to successfully work together and plan on working together again on the next project.

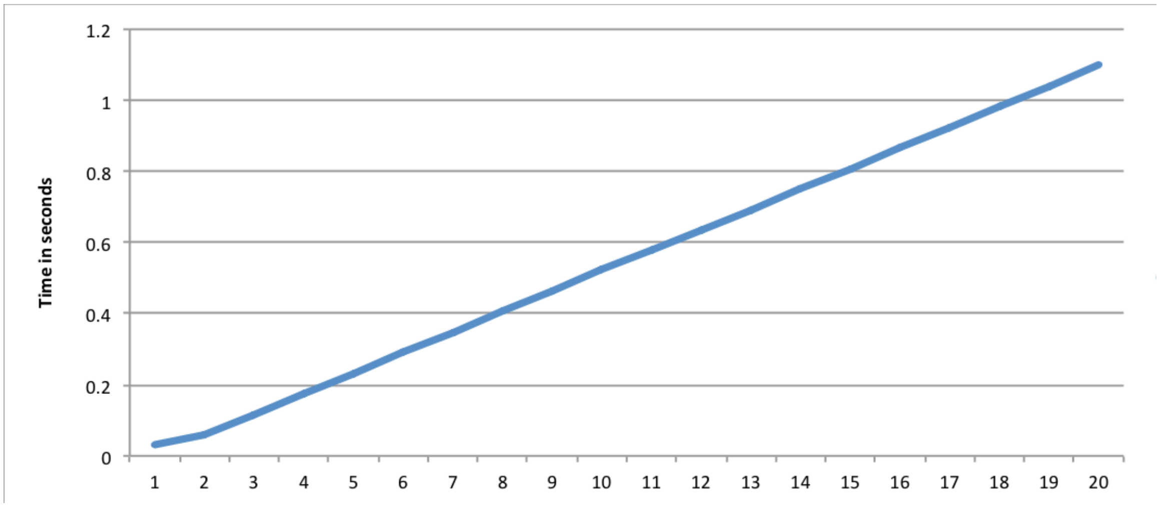
Our implementation would have differed in the efficiency and well as the variety of methods that we could have called. It definitely would have been more efficient as the methods have been written and tested to find the most efficient code possible. As far as program running time, there are pros and cons. On one had we only wrote the methods that were necessary for our small scope so there was less code to have to run through. On the other hand, because we ourselves wrote most of the code, there are definitely more efficient ways to write it that we are not aware of or not capable of. With respect to program development time, it would have been much more efficient to use a Java list as all the methods are already written and we would not have had to spend as much time writing testing code and debugging all of our own methods.

For our contains method, we were expecting the complexity to be O(log(n)) as we were using a binary search, and didn't have to do extra work to sort, swap or remove and item. However, after plotting it, our growth rates did not match my prediction and appear to be more linear and of complexity O(n) than of the shape log(n). I attached a PDF of our plot of the running time.

In the worst case scenario, our big O complexity would be O(n);

We spent about 15 hours total.

Kara Douville
u0305506



The x-axis is number of books times 10^6