CS2420 Assignment03 Analysis

ANDREW WORLEY / U0651238
University of Utah

1. Brian Park was my partner for this assignment, and I submitted the source code.

2. Initially we switched roles evenly taking turns driving and observing. Towards the middle of the assignment we started to encounter problems with our program and realized that our styles differed. This led to some competition to get our ideas implemented. However, this realization led to practices that led me to grow and develop better communication skills. At the end and completion of the assignment we switched roles more frequently and were "on the same page" more often than before. I prefer switching roles more often than not, and this is largely due to giving me and my partner more time to demonstrate our perspective to the assignment tasks.

3. I would work with Brian again. He was motivated and willing to put in the hours required to complete each task of this assignment. We were able to attend TA help hours and conduct our work together. However, the lessons learned motivates me to continue switching partners to gain a broader perspective of partner programming.

4. The java ArrayList, for example, comes with add and remove, auto grow and shrink capabilities. However, it allows duplicates and is not sorted until called into Collections. Using our generic collection class sorts elements as they are inserted.

5. I expected the Big-O behavior to be logarithmic because we implemented a binary search for elements in the collection. This search method involves halving when running iterations through the collection.
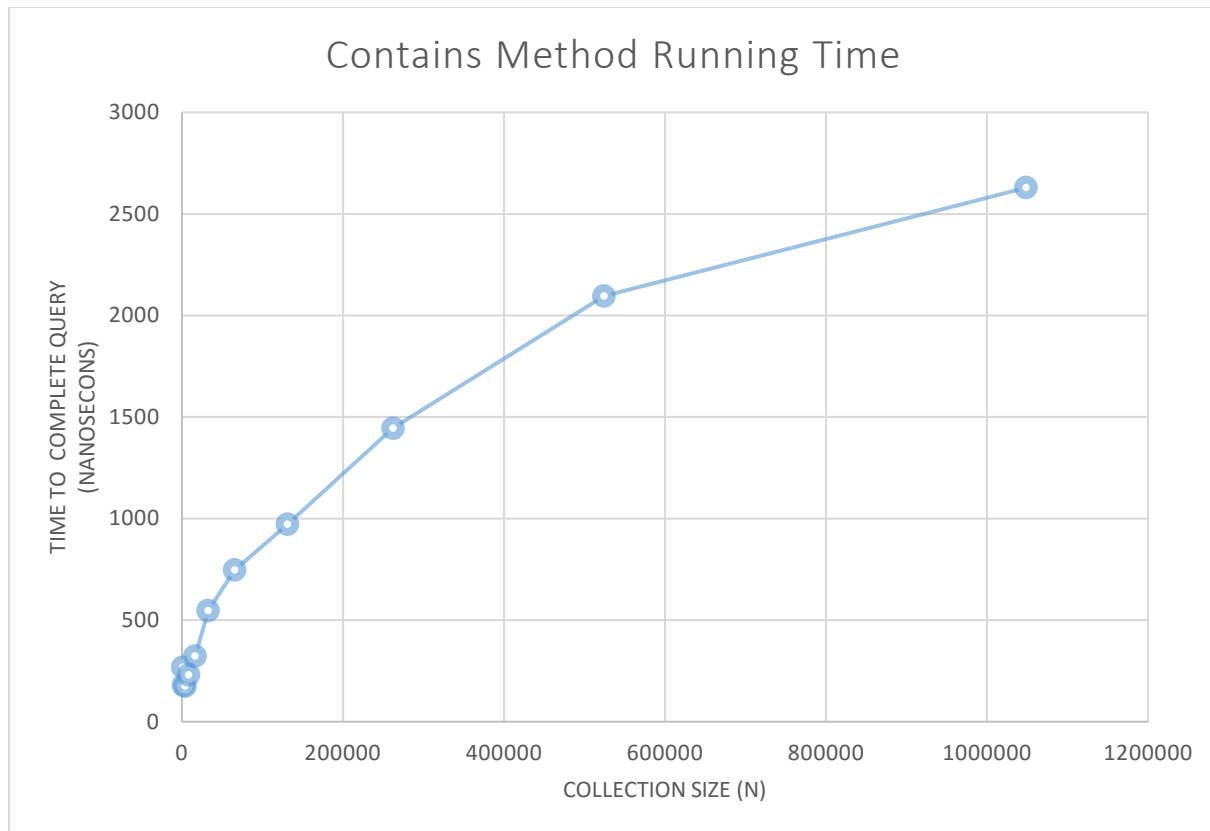
CS2420 Assignment03 Analysis

Figure 1. Contains Run time

6. Analyzing the graph in figure 1 demonstrates, as the collection grows, the time to find an element increases at a logarithmic rate. This is expected due to the use of a binary search implemented in the collection contains method.
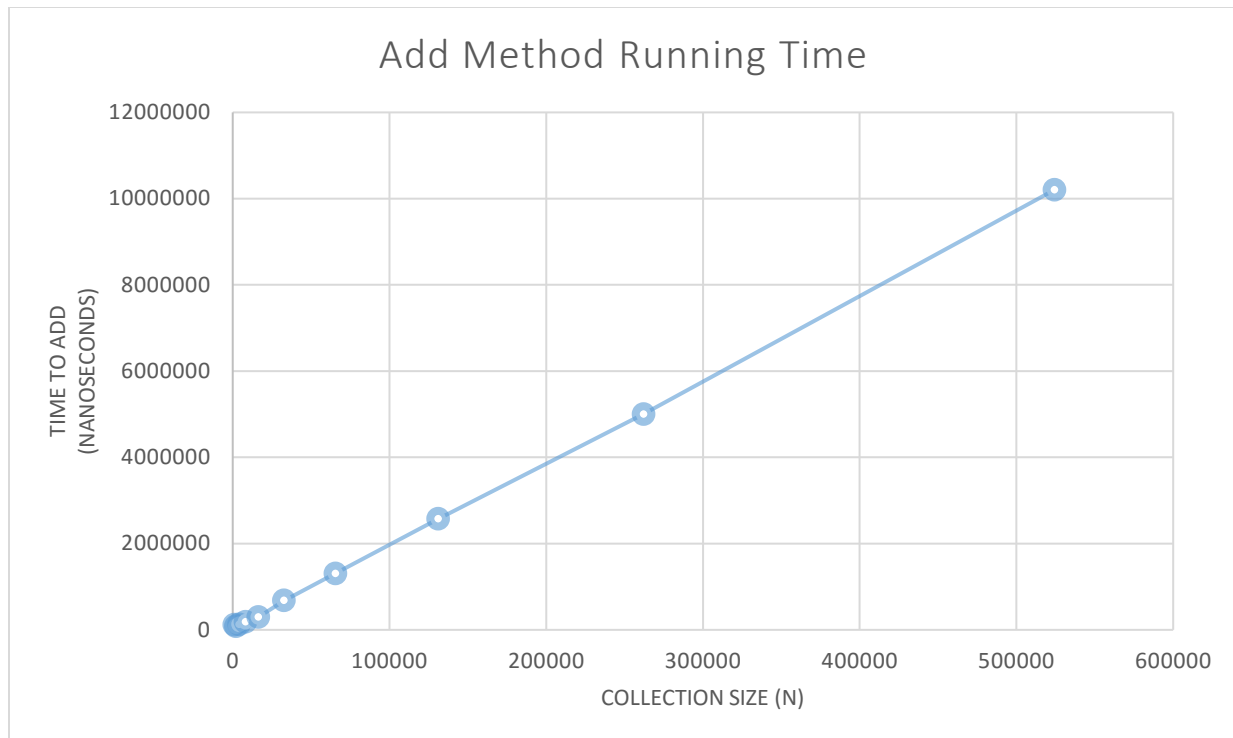
CS2420 Assignment03 Analysis

Figure 2. Add method run time

7. Figure 2. Represents the overall complexity of the add method as the collection grows. It follows a linear model and is largely due to the time taken to grow the collection or add elements to the 0 index.

8. We spent 15-20 hours on this assignment due to rewriting sections of code that we did not feel were the most optimized. The binary search method and add method went through many alterations. The binary search refinement increased our programs efficiency. Overall, all the debates and arguments that were involved in assignment led to better organized code and optimization.