Assignment 3

Analysis

-By Amit Athani

uNID: u0914565

1. My Programming Partner

My programing partner is Natalie Cottrill-Yavornitzky. She submitted the code.

2. Pair Programming/Role Switching

We switched roles every 20 minutes. I preferred switching less often. I as a programmer like coding for long stretches, I believe switching roles disrupts one's concentration and flow of thought. Hence a programmer is more likely to make a mistake if he disrupts his though process. This leads to sloppiness and poor coding, which is the primary cause of loopholes and bugs in the program.

3. Evaluating My Programming Partner

Natalie is a great programmer, she brought in a lot of experience and technical knowledge regarding diagnoses and testing of code and how to break the code down by testing. So that, these cracks in the code can be fixed, to create a much more resilient program. Working with Natalie was a learning experience for me. It strengthened my strategy for testing the code, which is a very important aspect of programming. Her programming logic was great, hence we could spend more time on improving the program rather than explaining the logistics of the code. She also helped me improvise my design for commenting and formatting, which made the code much more readable.

4. Java List vs Arrays

In the current assignment, we had to utilize an array to store the elements for our set. By extending the use of arrays, we were able to grasp the backend functioning and design of Lists, which are core concepts of programming. It taught us fundamentals of programming such as utilization of generics, binary search and other interfaces which form an essential part of any Java List. However, if we had utilized a pre-defined List (Such as an ArrayList or LinkedList), it would have offered the user much more flexibility by offering a wide collection of methods which were beyond the scope of the current assignment, while at the same time not compromise on efficiency and speed.

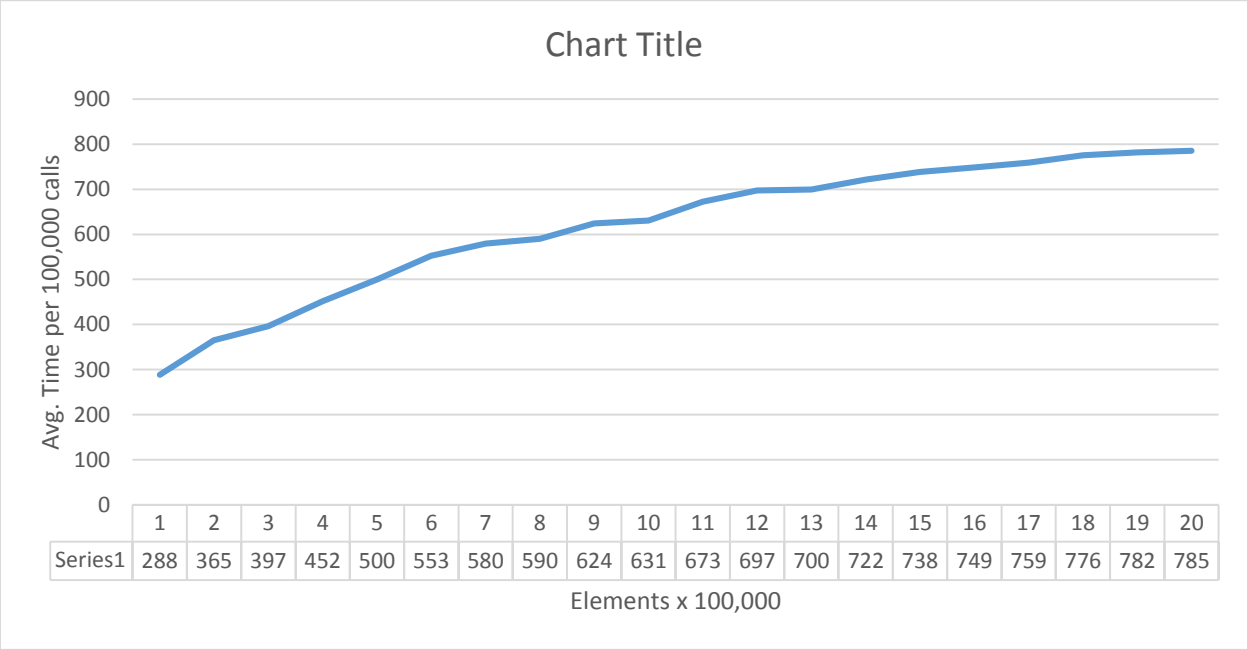
In terms of performance and speed, ArrayList are comparable to regular Arrays. This is due to the efficient coding structure and compilation of these data structures which results in better performance and versatility.

The greatest benefit of using pre-defined data structures is the availability of pre-defined methods such as sorting, adding and removal which will considerably reduce the time consumed on program development while at the same time improve the efficiency of the program.

5. Big Oh of Contains Method

The MySortedSet's, **`contains`** method utilizes binary search to locate the parameter in the set. Binary search works on the principle of dividing the array into two halves. This results in improved efficiency compared to other linear methods. Since, Binary Search works on the principle of "divide and conquer", hence I expect the Contains method to behave in the order of log N. The logarithmic properties of the method **`contains`** can be verified by graph plotted below, which within the limits of experimental error, exhibits all the characteristic of a logarithmic function curve.

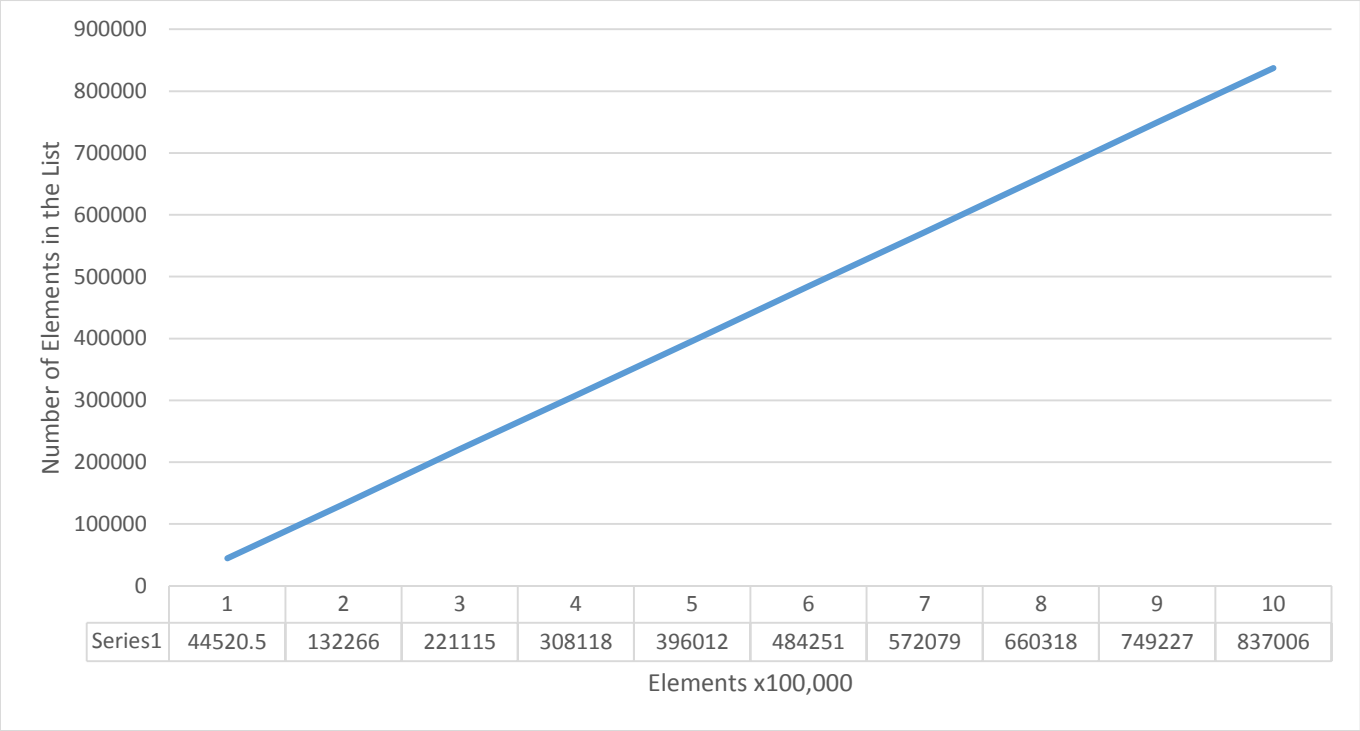
6. Plot for **`Contains`** Method



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Series1	288	365	397	452	500	553	580	590	624	631	673	697	700	722	738	749	759	776	782	785

Within the limit of experimental analysis, the plot for the graph matches the expected order for contains method which utilizes the concept of Binary Search to work in the order of log N.

7. Big-Oh of **`Add`** Method



The **`add`** method utilizes binary search to find the location to add the element, and the element are iterated through to shift them ahead in the list to create room to add the new object passed to the **`add`** method. On analyzing this, we predicted the method to be the Order of log N + N. N being the dominant term, we can simplify to say that **`add`** method is of the Order N or Linear. This can be verified by observing the graph, I plotted on observations based on the add method, which exhibits all the characteristics of a linear order program.

8. Time Spent

We spent around 9-10 hours working on this assignment. Out of which we spent about one half of an hour on the design, semantics and scheduling of our project, another 4-5 hours was spent on coding the assignment, the rest of about 5 hours were spent on testing the code. I believe, we also consumed a lot of time analyzing the requirements of the project and what the methods expected us to check and return. A large portion of our time was also spent on figuring out how to properly implement the Iterator and figure out how to effectively use the comparator provided to us in the constructor.