Tianyu Wang
U0911557
Partner Cody Ngo

When you are satisfied that your program is correct, write a brief analysis document. The analysis document is 30% of your Assignment 3 grade. Ensure that your analysis document addresses the following.

1. Who is your programming partner? Which of you submitted the source code of your program?

My partner is Cody Ngo, and my partner will submit the source codes.

2. How often did you and your programming partner switch roles? Would you have preferred to switch less/more often? Why or why not?

We switch a lot. Because both of us have some good ideas about the code. So for different parts we always change the ideas and change the roles.

3. Evaluate your programming partner. Do you plan to work with this person again?

My partner is really good. He has a good understanding about java, and knowing what should we do. So I plan to work with him again.

4. If you had backed the sorted set with a Java List instead of a basic array, summarize the main points in which your implementation would have differed. Do you expect that using a Java List would have more or less efficient and why? (Consider efficiency both in running time and in program development time.)

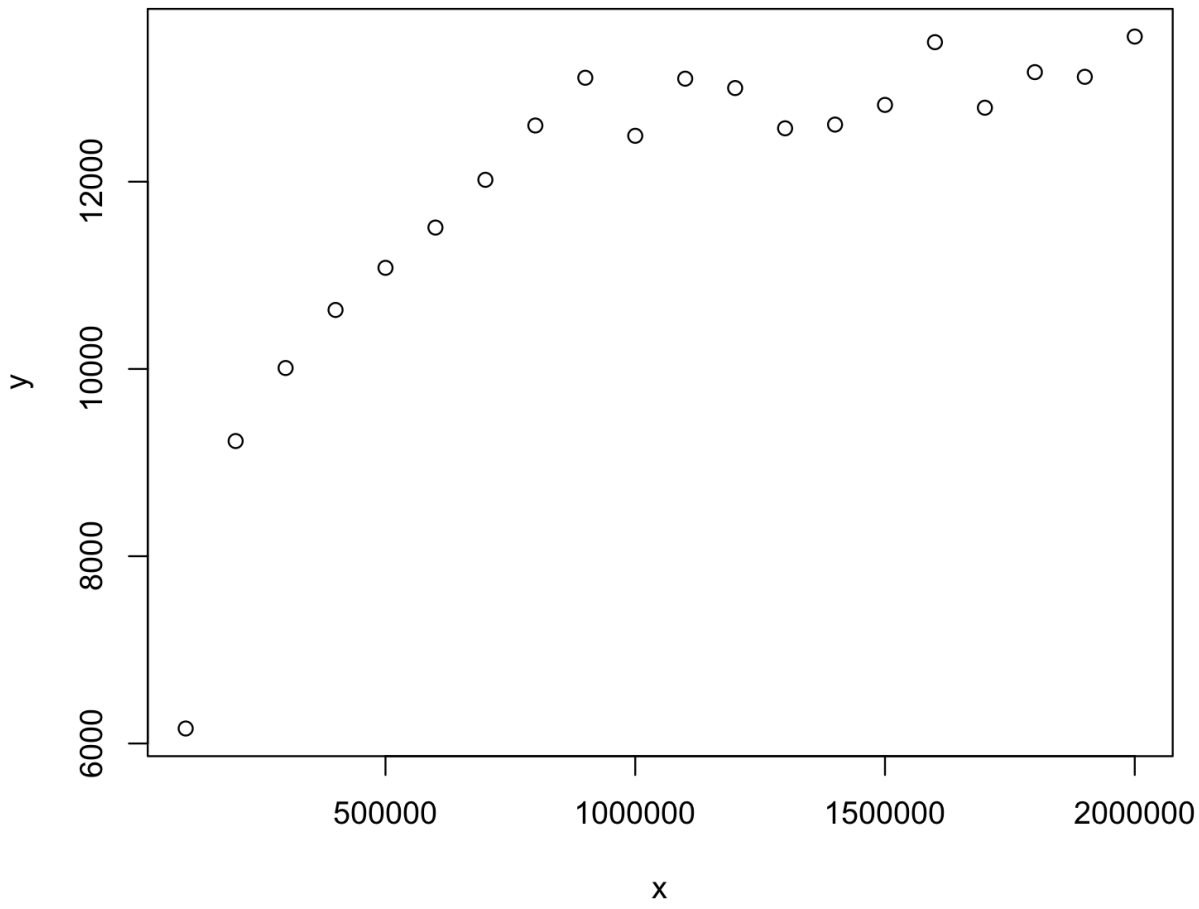
In my opinion, I think using Java List will be more efficient. For example, the basic array must fix the

length. So if the length is full, it will create a new one, which have a bigger size and copy all the code to it. Therefore it will spend more time. In addition, if we frequently add elements to the beginning of the List, using LinkedList. It will spend a constant time in LinkedList, however, it will spend linear time or even more time in basic array.(at least the O(logN)). Therefore, I believe java List will save more time.

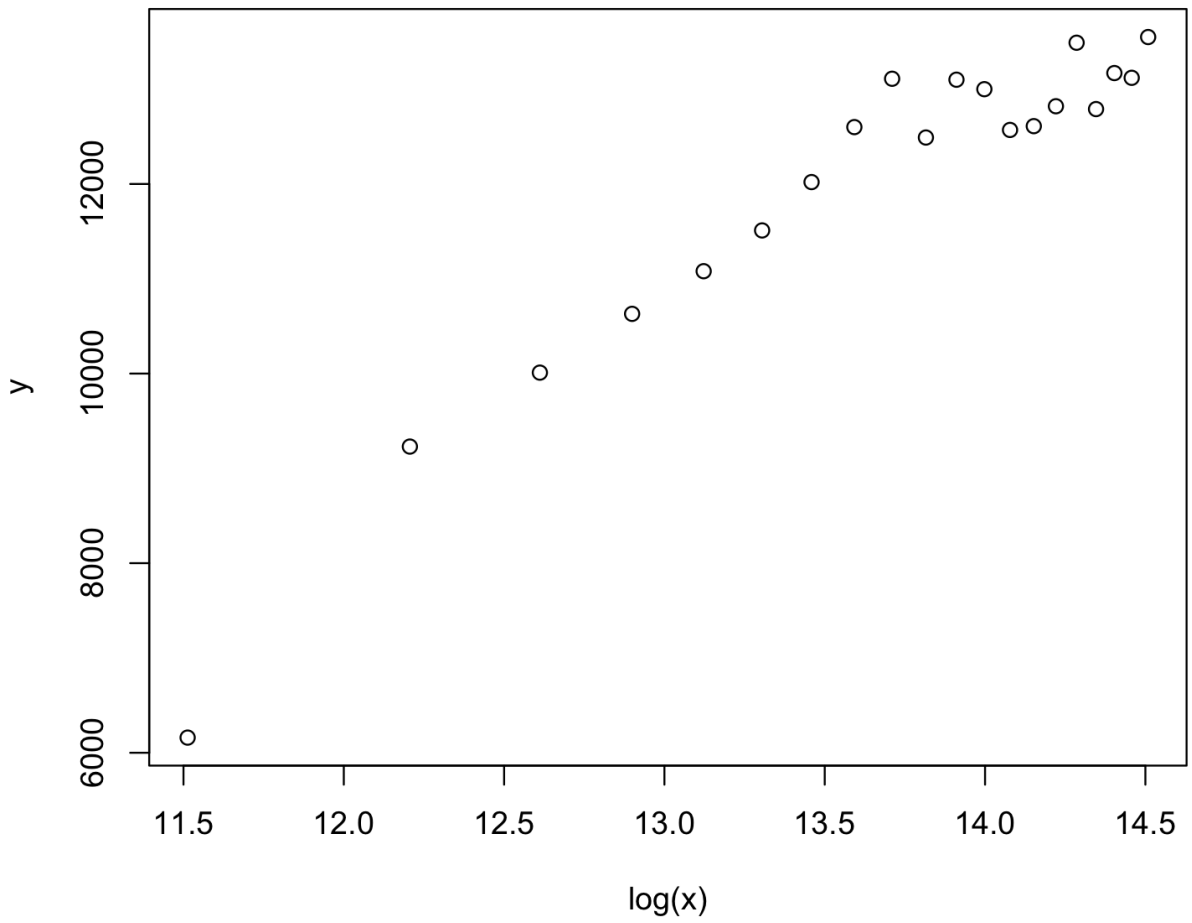
5. What do you expect the Big-O behavior of MySortedSet's contains method to be and why?

I prefer it to be O(logN) Because at that time when the array is large enough, it will save more time.

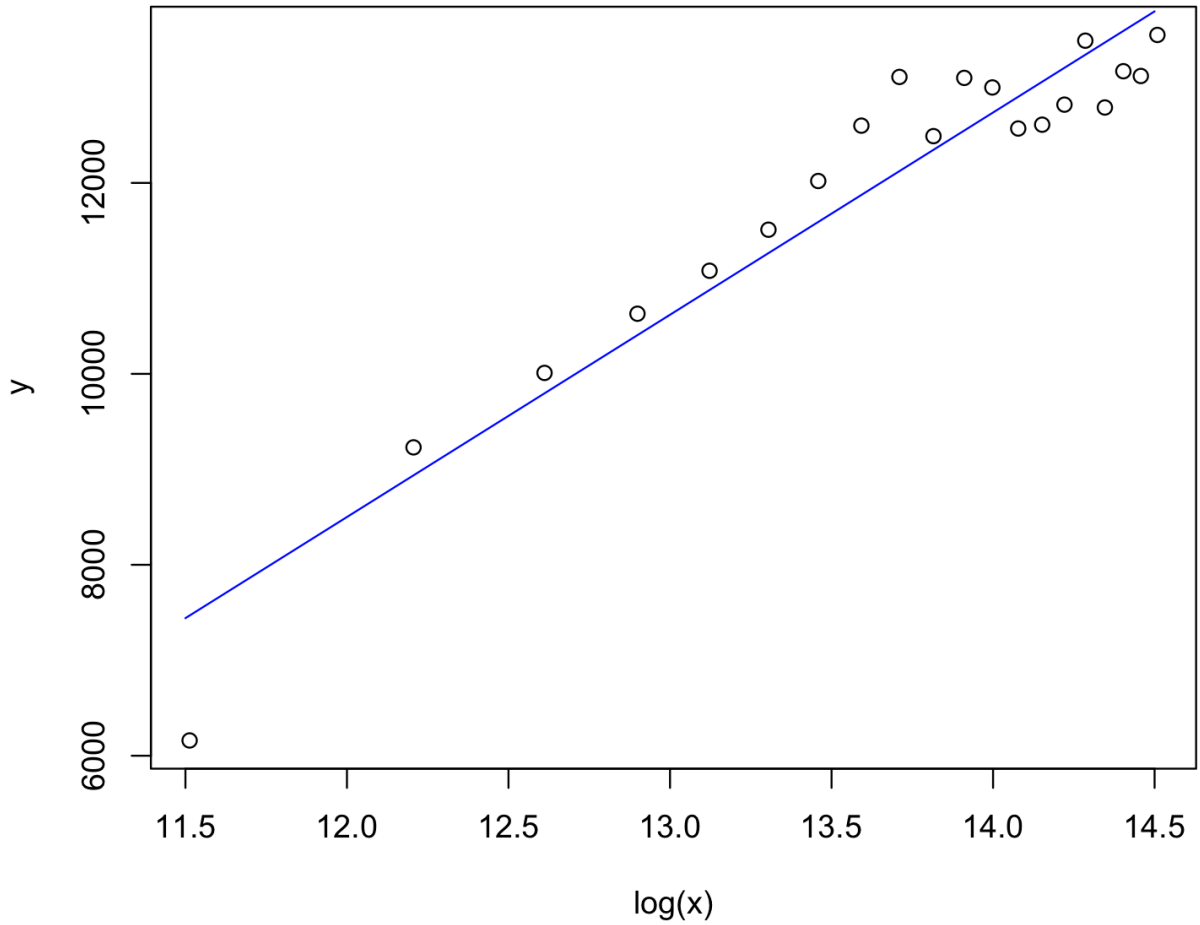
6. Plot the running time of MySortedSet's contains method for sets of sizes 100000 to 2000000 by steps of 100000. Use the timing techniques demonstrated in Lab 1. Be sure to choose a large enough value of timesToLoop to get a reasonable average of running times. Include your plot in your analysis document. Does the growth rate of these running times match the Big-oh behavior you predicted in question 5?



This is the figure about y~x if we draw another figure about the y ~ log(x) we can see



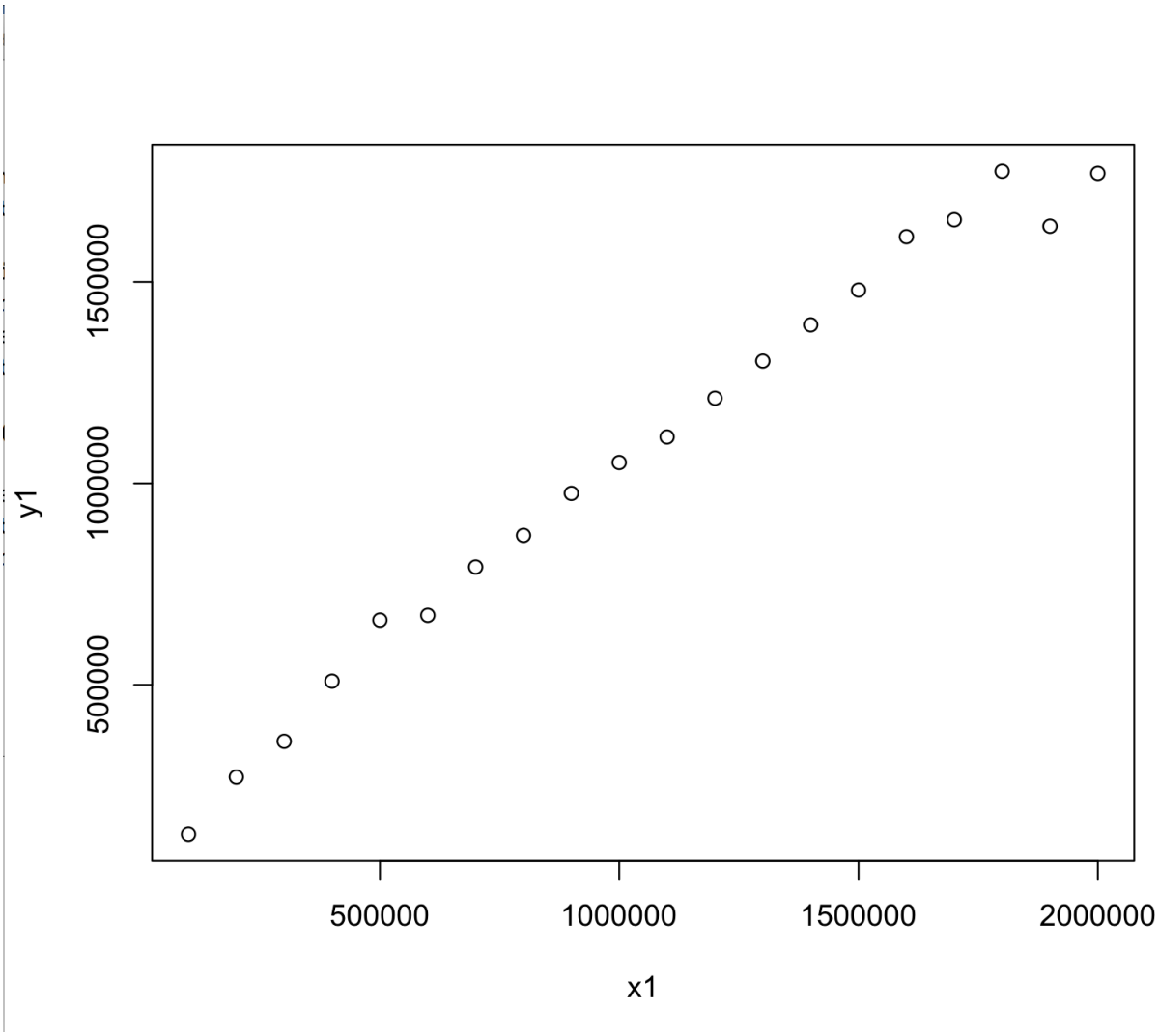
Using this data, we can try to find a linear-regression. If we can find a line cross most of the data, then it's linear. Fortunately, we can find it.



Therefore this search method has a high probability about the O(log(N))

7. Consider your add method. For an element not already contained in the set, how long does it take to locate the correct position at which to insert the element? Create a plot of running times. Pay close attention to the problem size for which you are collecting running times. Beware that if you simply add N items, the size of the sorted set is always changing. A good strategy is to fill a sorted set with N items and time how long it takes to add one additional item. To do this repeatedly (i.e., timesToLoop), remove the item and add it again, being careful not to include the time required to call remove() in your total. In the worst-case, how much time does it take to locate the position to add an element (give your answer using Big-oh)?

It seems not so outstanding, but in theory, it should be



O(logN). Because in the worst-case, in total m number to find an element needs log(M) (base on 2) + 1. So it will spend O(log(N))

8. How many hours did you spend on this assignment?

9 hours