Name: Jin HE

UID: u0870268

Class: CS2420-007

Analysis For Assignment3

1. My partner's name is Qixiang Chao, and he will submit this assignment.

2. We always switch role during the group working. Actually, I was mainly play the role as a driver in this assignment, but my partner would be a driver and I would be a navigator. Therefore, I could think out some useful points in order to overcome some intractable problems, and guide my partner to implement my ideas.

3. My partner did a really nice job in this assignment. Both us thought this project is harder than the last two. So, it expend us a plenty of time for the code writing and debug. Currently, our program got three times faster than the previous one.

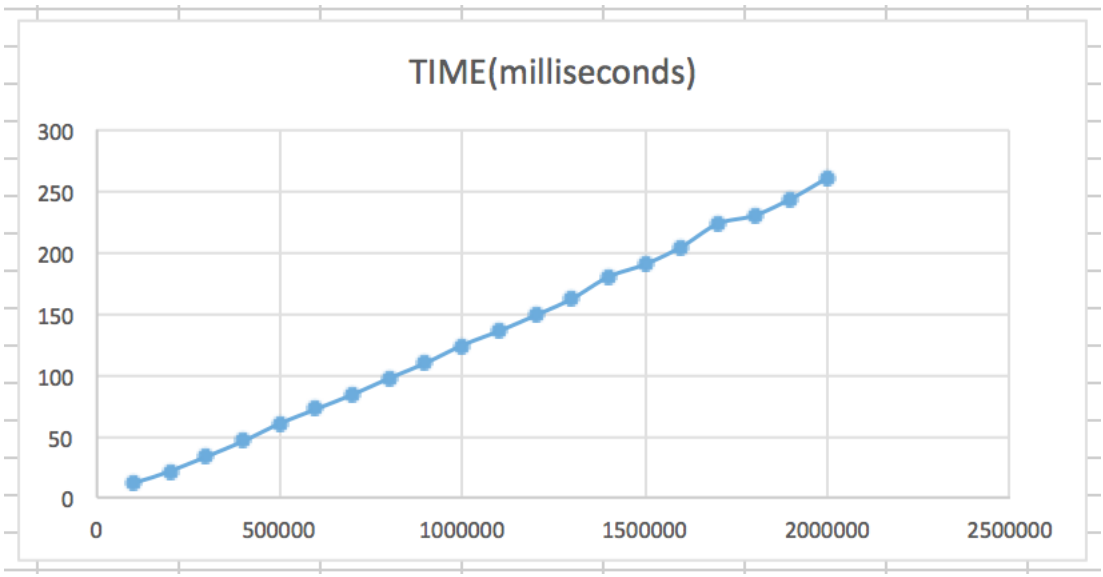
4. In order to answer this question, I compared our program (MySortedSet<E>) by using the "ArrayList" and "LinkedList". I added 100000 elements into these three collections. It took only 16 milliseconds when added elements into the "ArrayList" . However, it took 19 milliseconds to add elements into the "LinkedList" and 33 milliseconds into the

"MySortedSet". Nevertheless, I know that the "ArrayList" and "MySortedSet" need to cost a long time to enlarge the size (size*2) of the array when we want to try to add elements with a large scale. On the other hand, it will also cost vast times on copy every element from the original array to a new array constantly when the size is larger than the original size. The "LinkedList" uses the link node instead of array to implement its function. So, it doesn't need to enlarge its size in order to add more elements. Logically speaking, the "LinkedList" is working faster than the "ArrayList". In general, I prefer using "MySortedSet" if I need a quicker speed for handling large scale elements insert into the set and sort the elements meanwhile. It will help me to save times on writing the extra code to take a sort of elements in the "ArrayList" and improve the efficiency of my program accomplishment. And the "LinkedList" need to take a long time to check every node inside it. It means that needs a for-loop if we really want to use it to implement the sorting function inside it(the Big-0 complexity is N). This character also decides that the "LinkedList" is not suitable for sorting the elements. I prefer using the "ArrayList" for the collection if I don't need to sort elements because of its high speed.

5. The complexity of contains method is O (log N) because of it uses the binary search. It

means the loop just needs to check half of the element in the array to know whether the element have already contained in the array.

6.

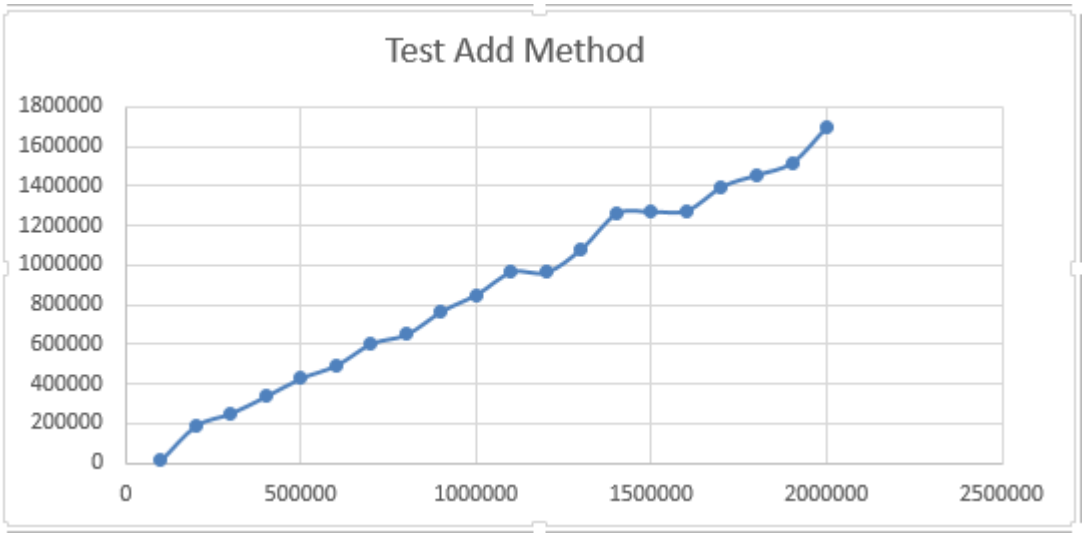


This is the graph that I made after testing for contains method from 100000 to 2000000. And actually it's approximate a linear graph. The x-axis represent for the elements total of test. And the y-axis represents for the time that calculated as nanoseconds. Actually I expected a log (N) graph for the contains method, but I don't know why I just get a linear graph. Maybe the total of elements is not enough.

7. This graph is used to test the add method in the "MySortedSet" class, I tried to add different random integer to 20 different sizes "MySortedSets". The result is a linear graph. And in my code, the worst case for the add method is O(N). I got pretty small time many

times during the process of test. I think it must meet the best case. The worst case is

Big-O=O(N).



8. For this assignment, I have cost 15 hours on it.