Austn Chen
Analysis-doc 3
CS2420

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

Stephan Stankovic. Stephan submitted the assignment

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 did most of the coding together, so each of us shared all parts equally.

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

Stephan was a great partner. He gave ideas and even wrote code even while he was sick. We split most of the work equally so we had just as much to work on. Both of us had work so finding time was tough. We still planned accordingly to our schedule and worked on it as much as we could together. I plan to work with this partner 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.)

I think it would be more efficient because we would already have most of the methods written for us. The run times would be fast because it will set be coded inside java.

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

I think it would be N^2 because it has to loop through the array and find if the element we’re looking for is available or not.

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?

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)?

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

We spent about 10 hours on the assignment.