Colton Haacke
u0935270

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

My partner was Cole Cruz. I submitted the code.

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

My partner and I switched roles every other method. I liked how it worked out and prefer the way we did things to switching more or less often. It let us do a good amount of work and let us focus on the method without having to switch too often.

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

My partner did a good job. He was easy to work with, knew what he was doing and did his fair share of the work. He has chosen another partner to work with for the next project, but I would be glad to work with him again if he wanted to.

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.)

If we used a Java List instead of a basic array, then we would not have had to keep track of how many elements were in the array at any time, and could have just used the length of the array instead. There also would be no need for a grow method for the array. I feel that a Java List would be the same efficiency in running time, but more efficient in program development time.

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

I expect the Big-O behavior of the contain method to be O=log(N), because we kept cutting the area that could be searched in half with every loop.

6. Plot the running time of BinarySearchSet's contains method, using the timing techniques demonstrated in Lab 2. Be sure to use a decent iteration count 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., iteration count), 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?

I spent about 12 hours working on this assigment.