

1. My partner is Hannah Hollberg. She turned in the source code for our program.
2. We switched roles twice. In the future I would probably prefer to switch more. This will give us both more experience in each roles at different stages of development.
3. Again, Hannah was great to work with. We work well together, and I feel that we contribute equally to the finished product. I look forward to working with her again.
4. A java list would already have many of the methods we implemented in MySortedSet, such as add, clear, contains, containsAll, isEmpty, iterator, remove, removeAll, size and toArray. This would have definitely been more efficient in program development time, but in many ways would have been less efficient in run time. For example, the contains and containsAll methods would not take advantage of binary search.
5. I expect the big-O behavior of the contains method to be $\log N$ because it implements a binary search, which cuts the set in half in each iteration.
6. See next page for plot. This plot look like it roughly agrees with $\log N$ behavior.
7. See next page for plot. In the worst case, it would take $N \log N$ operations to add an element. It would take $\log N$ operations to run the binary search and N operations to shift all of the elements down in order to make room for the new element. We found it very difficult to accurately calculate the run time for the add method because we were unable to subtract the amount of time it take to remove an element at a certain position. After speaking at length with Hitesh we came to a somewhat rough estimation by running both run and remove in the loop and dividing the time by 2. This works because adding and removing an element from the same position take roughly the same number of operations.
8. This assignment took us approximately 8 hours to complete.

