

- 1: Jeanette Arteaga was my partner for this assignment. I (Aaron Pabst) submitted the code.
- 2: We only switched roles once or twice. Most of the work for this assignment involved thinking and planning, not coding, so it did not matter who was at the wheel.
- 3: My programming partner did an excellent job of being engaged in the assignment and contributed a lot. I will work with her again.
- 4: We implemented the sorted set with an array. I would speculate that using a list would be much less efficient seeing as a list is a collection itself. However, lists provide for more flexibility and may have been easier to code.
- 5: Since contains uses binary search to locate elements, the contains method has complexity $O(\log n)$.
- 6: We had difficulties computing the running time of contains for large N. We could not get more than a few runs to execute in a reasonable amount of time. We could determine that 100000 took 27372 ns and 200000 29935 ns. Anything that and we could not get a successful test. We would find more information on timing programs useful. This does fit a logarithmic complexity as the time difference between the runs was very small.
- 7: The add method uses almost the same algorithm as contains, except that it has to sequentially loop through the set in order to add an element, meaning that it has linear complexity. The worst case would be if the element was at one of the outer extremes of the set. The best case would be if the element was already present as we could then exit the function as soon as the element was found. There would also be no need for a scratch array in that case. We can observe through our timing tests on contains that the size of N greatly affects running time for add. (For that test we added 100,000's of random elements to a set.)
- 8: We spent approximately 9 hours on the assignment. It would have been much less if we had gone through a proper planning phase.