

### Generic Sorting Analysis

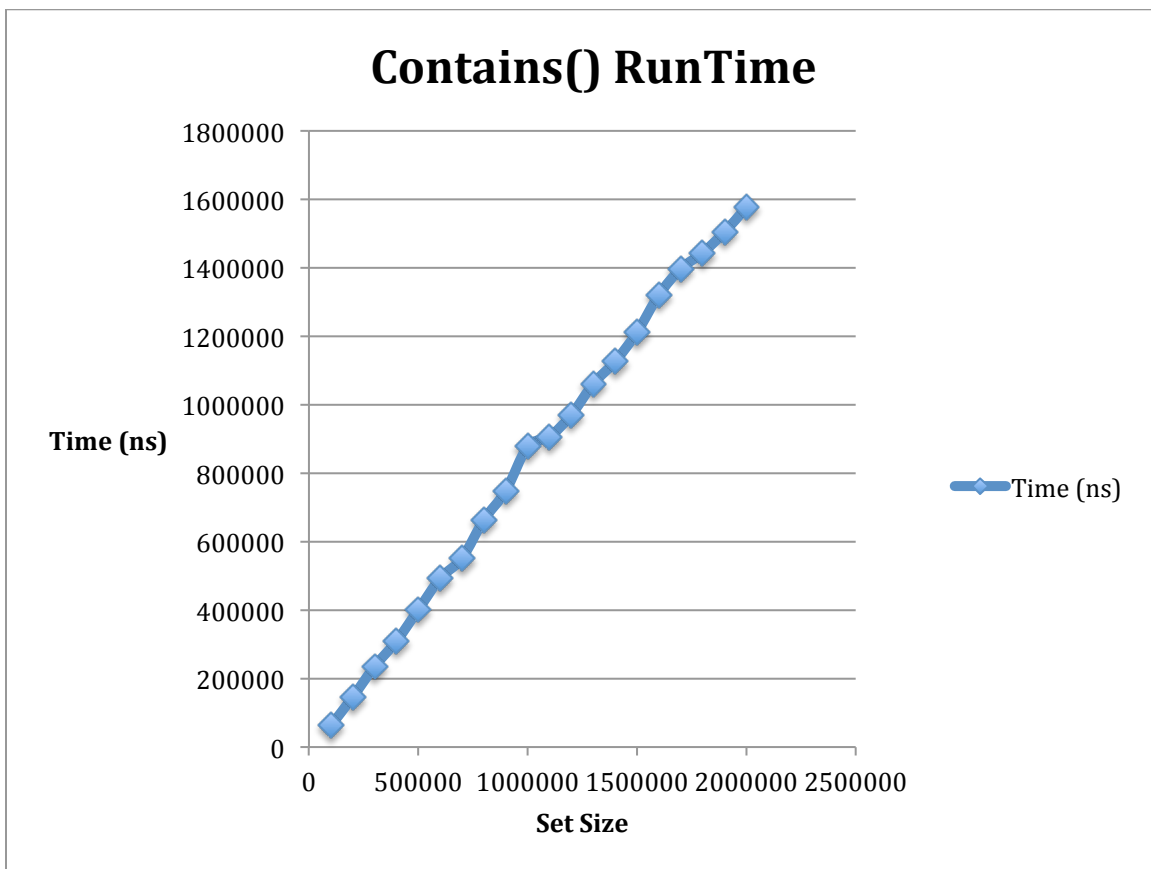
For the generic sorting programming assignment, my partner was Tony Chow and this time I will be submitting the source code for this program. Coding this assignment, we switched navigator and driver frequently, but usually only after times of frustration. Working with Tony has improved because we are beginning to work more cohesively and I am able to manage my time much better. I definitely feel like I am becoming a better coder even though I was unable to turn this assignment in on time. Tony is a great partner and helps motivate me to improve.

If we had used a Java List instead of a basic array, we would no longer need a grow function and this would allow us to have a less extensive code. Although as stated in class, simpler functions usually have a greater Big-O complexity and would increase the running time of the program. It would take much longer to bring up the list/set as the program would need to allocate twice the memory needed for a basic array so that items can be added quickly. A list that isn't receiving any additional values would still use up the same amount of processing.

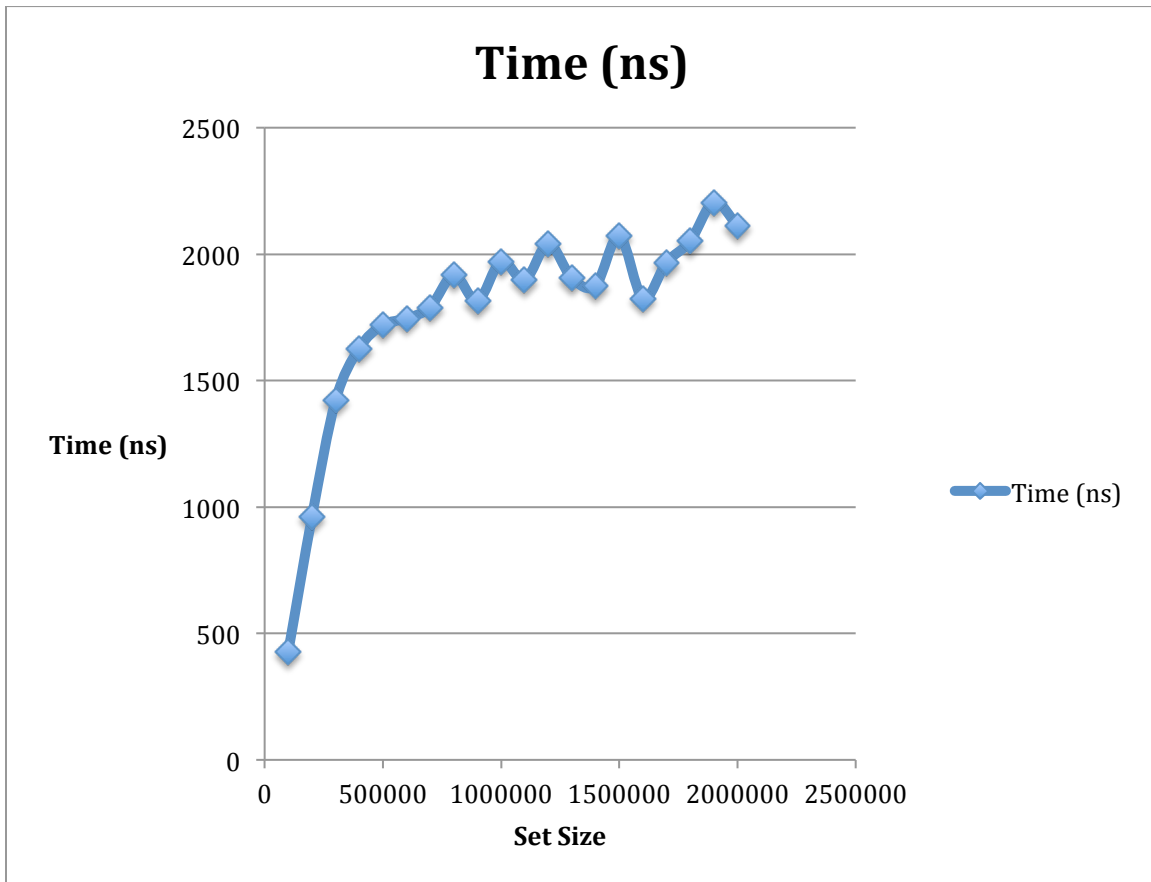
On the flip side, an array list would've allowed us to code the program much quicker and would've made the programming process more efficient. Less lines code and having to worry about going out of bounds are two examples that would benefit the programmer. Only if it had been a large indexing of multiple bits of data, array list would've been more helpful, but with a smaller collection the basic array is faster.

My expected Big-O behavior for this program is roughly  $O(\log(N))$  due to the use of the binary search. It would need to go through the entire list and increase in relation with the length of the array but, with a more efficient method of search, in larger indexes the run time will plateau.

The `combine()` method had a linear slope so it did not match the logarithmic plot we had been anticipating. This could be attributed to errors in our code as well as inconsistencies in the java timing processes.



For the add() method, the run time was significantly less. In the worst case scenario, it would take a  $O(\log(N))$  complexity.



Approximately 18 hours were spent on the making of this project.