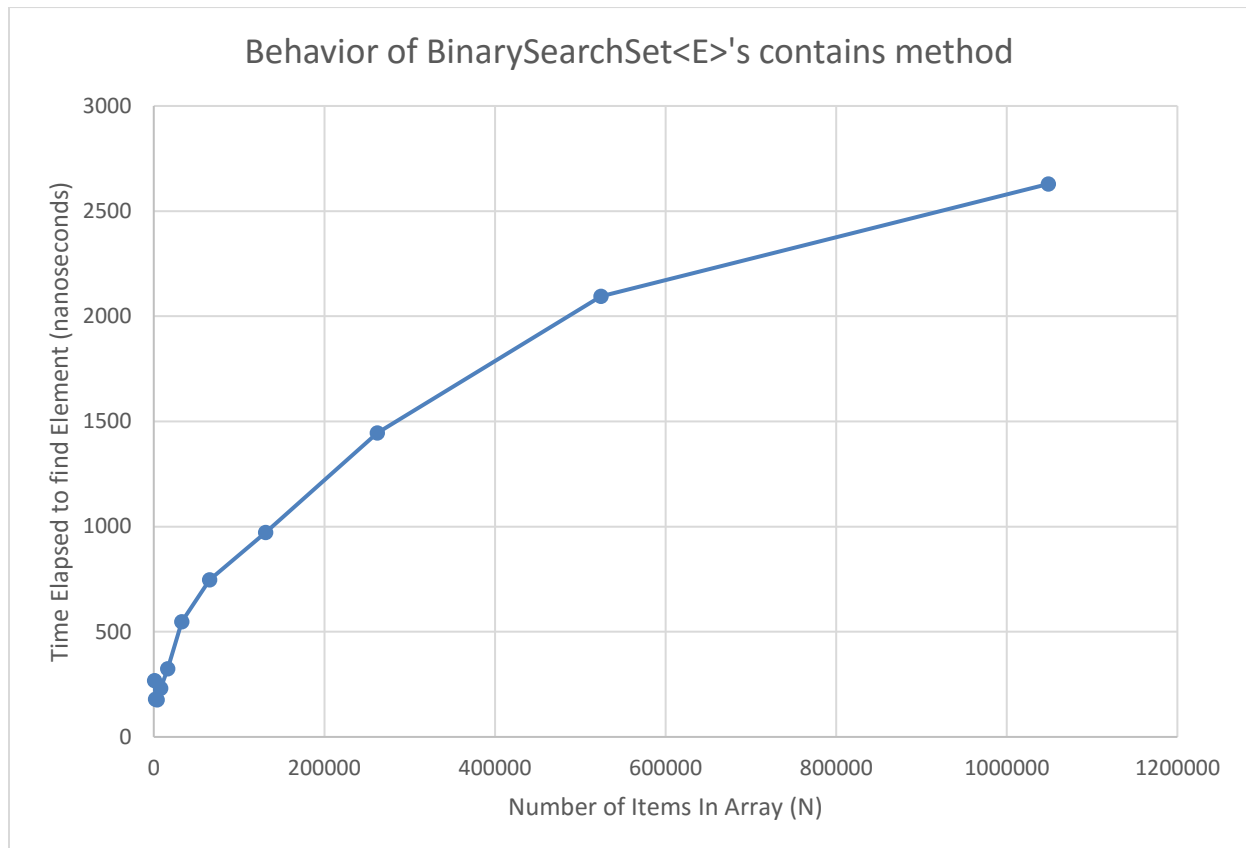
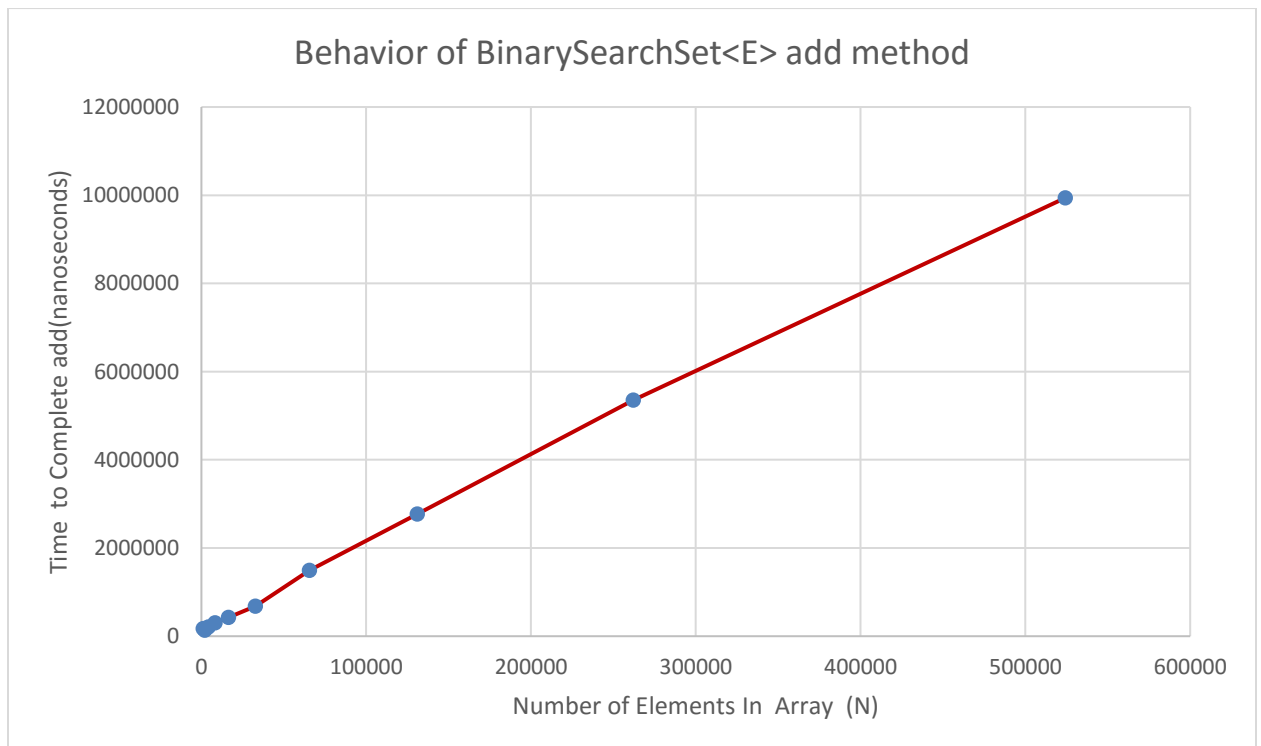


1. Andrew Worley is my partner. Andrew is submitting the source code for our assignment.
2. We switched fairly often at the beginning of the assignment but as we ran into harder problems it became frustrating to be switching so we ceased this practice. Eventually we found a rhythm of bug finding and taking turns implementing different features. I wish we had switched less at the beginning as this introduced quite a few bugs due to different coding styles and felt like we were rushing through code.
3. Andrew does a great job and although our coding styles are very different we have worked well together. I would work with him again as now we are already used to working together, and it would be difficult to adapt to a new partner compared to just getting better at working together. Then again I would be open to a different experience as we ran into several problems along the way resulting in time not well spent.
4. If we had backed our set with a sorted list then we would have not had to implement many features such as auto growing, however we would still have had to implement the contains method as well as other methods in order to perform binary search and keep the list in order. In the end it would have been slightly easier just because we would not have had to worry about the much overhead, so it would have save a significant amount of programming time, although would have been tricky and potentially less efficient.
5. Contains method should be of the order of $\log(N)$. This is the nature of the binary search as it is halving the next search size after each previous search. Because of this the efficiency as N goes to infinity is equal to $\log(N)$ with a base 2.

6. Yes our graph of contains with a high iteration count and a large N displays a graph that is similar to a type of log graph. As you can see below our graph reflects logarithmic growth for arrays containing a large number of element and tests that impellent large numbers of iterations.



7. For an element being added to a set, the time it takes to add depends directly on the number of items contained within the set. In other words as N gets larger, the time it takes to insert the item into the array get big at approximately a one to one rate. The big O behavior could be said to be that of N . As shown below this one to one growth results in the graph of elements vs time to be linear.



8. I spent approximately between 15 and 20 hours on this assignment however this was mostly due to not being efficient, finding frustrating bugs and getting used to coding in pairs. Over all it should have taken less time. In the future assignments should take less time as this was a good learning experience.