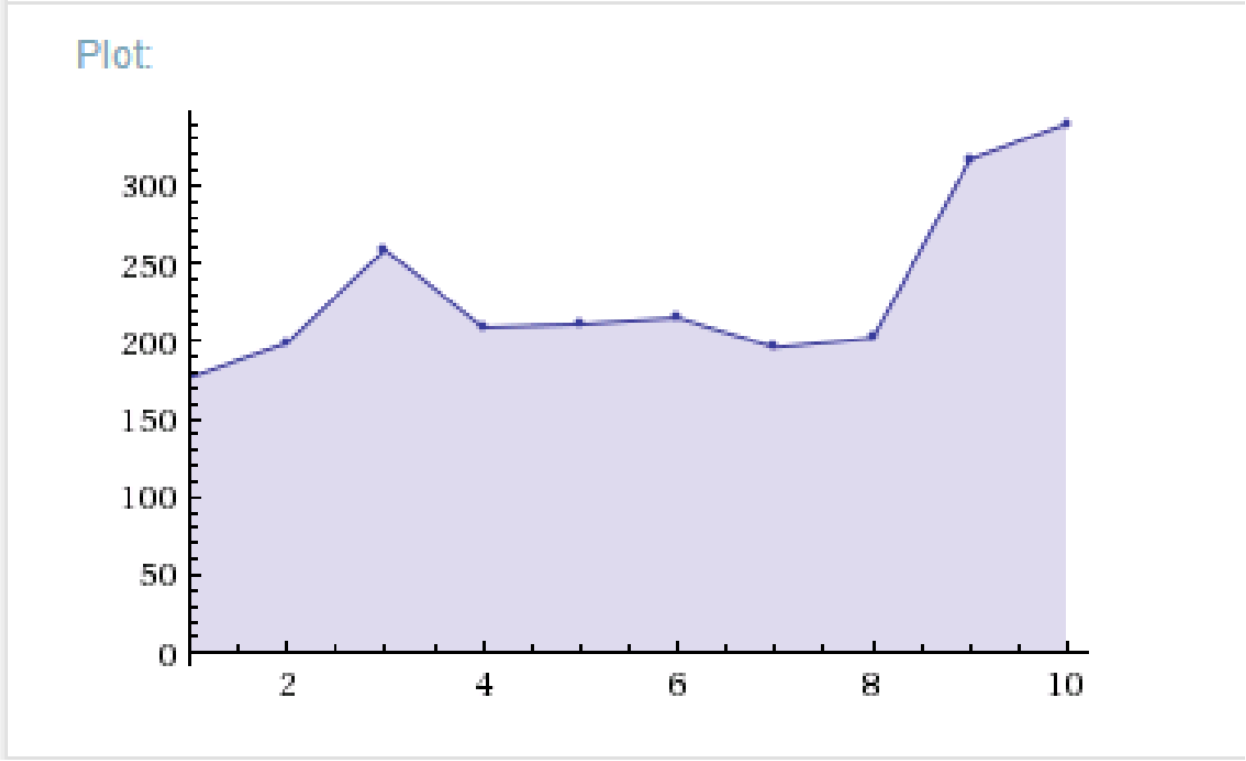
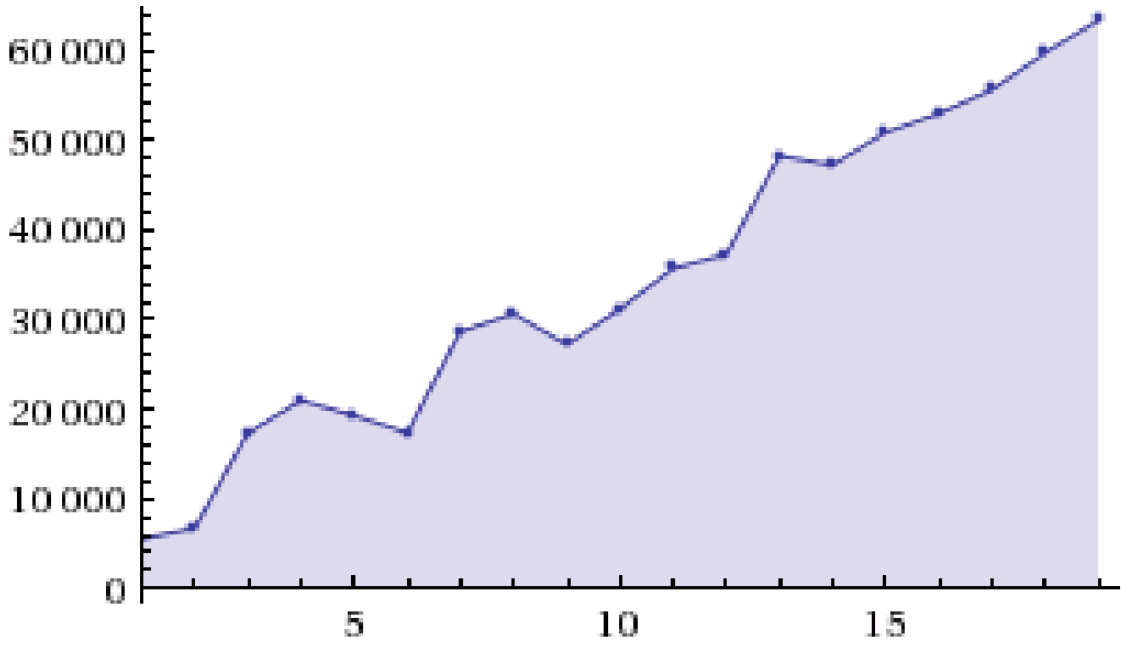
1. My programming partner is Derek Dixon. I submitted the source code.
2. We didn't switch roles very often. But we also didn't pay much attention to it. We just went with the flow of how the work was going best.
3. Great! Very good for bouncing ideas off of, and nice for talking to. I do plan to work with him again.
4. Since a Java List is just an interface, having our class implement it would change no actual code, and would just require us to add more methods. This would increase our development time, but nothing else.
5. The Big-O behavior of the contains method should be O(log(n)). It is a binary search, which halves the length of what needs to be searched each time.
6. The running time of the contains method (plot below— time (ns) vs hundred thousand elements in array) seems to reflect a largely logarithmic growth, although with some outliers. But this does match up with our expected behavior.



7. The add method— For an element not already in the set, the algorithm takes O(log(n)) to locate where to find the location the element belongs (being a binary search) (plot below). To actually add this element takes O(n) time, since not only does the location need to be determined, but the entire array needs to be recreated to allow for the new

element to be inserted properly into the other n elements. This matches up surprisingly well with the plot.

Plot:



8. We spent approximately ~10 hours throughout everything to do with the assignment.