1. My partner was Kory Hansen, and he submitted the code.
2. We switched roughly every half hour, but it was more dependent on which of us was feeling more clarity at a given time. The more confident person would navigate so that he could explain the logic to the driver while we went along. It was more fun, and made for better progress.
3. I do plan to work with Kory again. He's roughly on my level, so we learn things at the same time. Even though I'm slow and steady for him, I often keep him from making stupid mistakes in his code (I solve a lot of problems in our testing). And he pushes me along. I will be working with him again.
4. Implementation with a Java List would have been much easier, as it already contains many of the methods we had to implement, and handling generics also would have been largely taken care of. So programming time would have been drastically reduced. Running time would probably have been about the same (maybe a little better), since the Java List handles things in roughly the same manner our methods do.
5. O(log(n)) – .contains calls our comparing method, which in turn calls another compare – but these methods should be O(c)). .contains also runs a binary search to locate the position of the provided element, which is O(log(n)).
6. See Figure 1. The graph does not resemble a log(n) curve. We think because the elapsed time was so short, outside factors greatly skewed the results between tests.
7. See Figure 2. Since .add utilizes the binary search, the worst case for finding the insertion position is O(log(n)) time. The adding itself is O(nlog(n)) time since elements have to be moved over during insertion.
8. >14 hrs…

Figure 1.

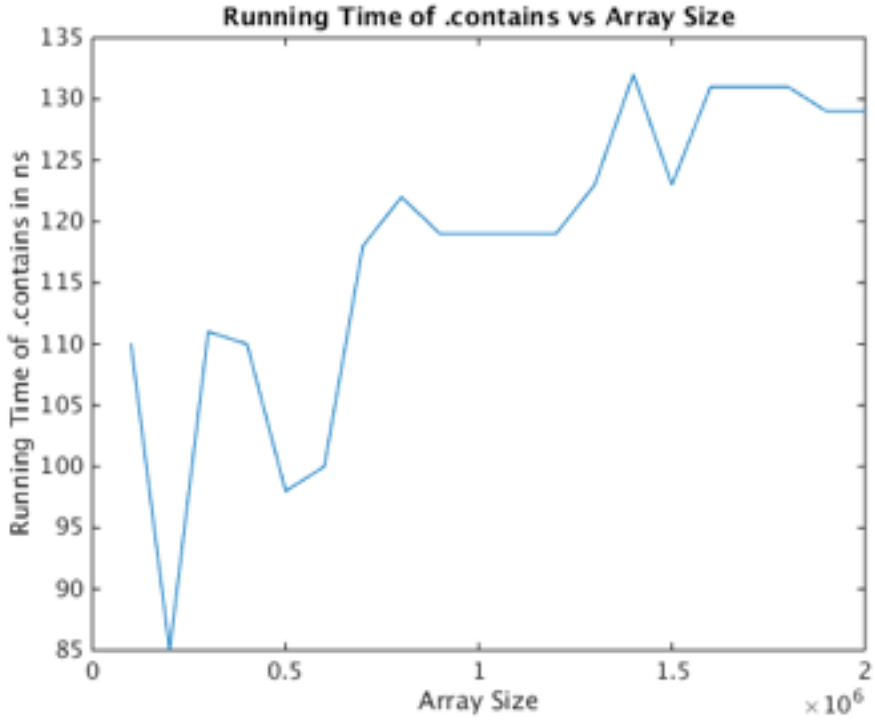

Figure 2.