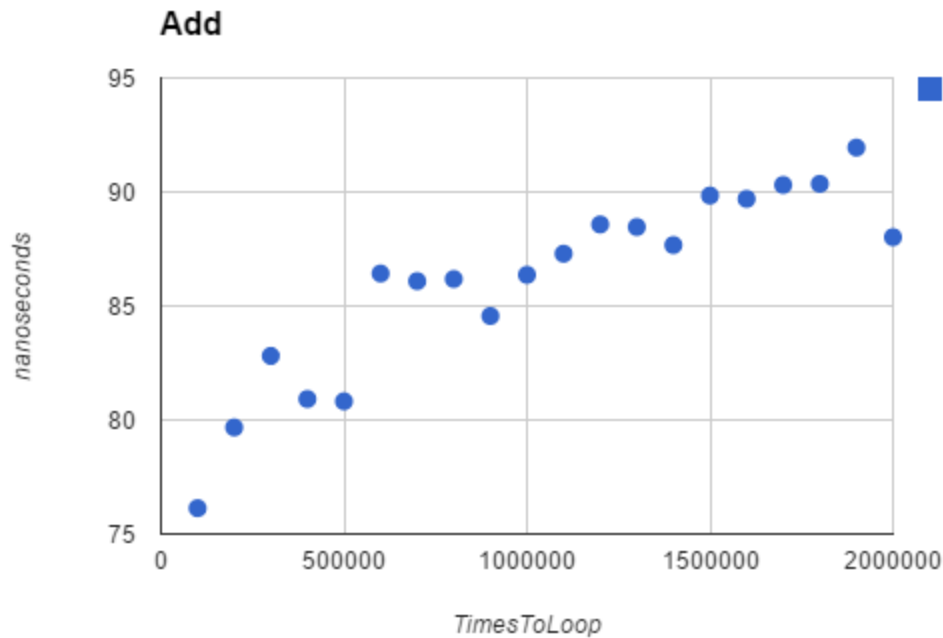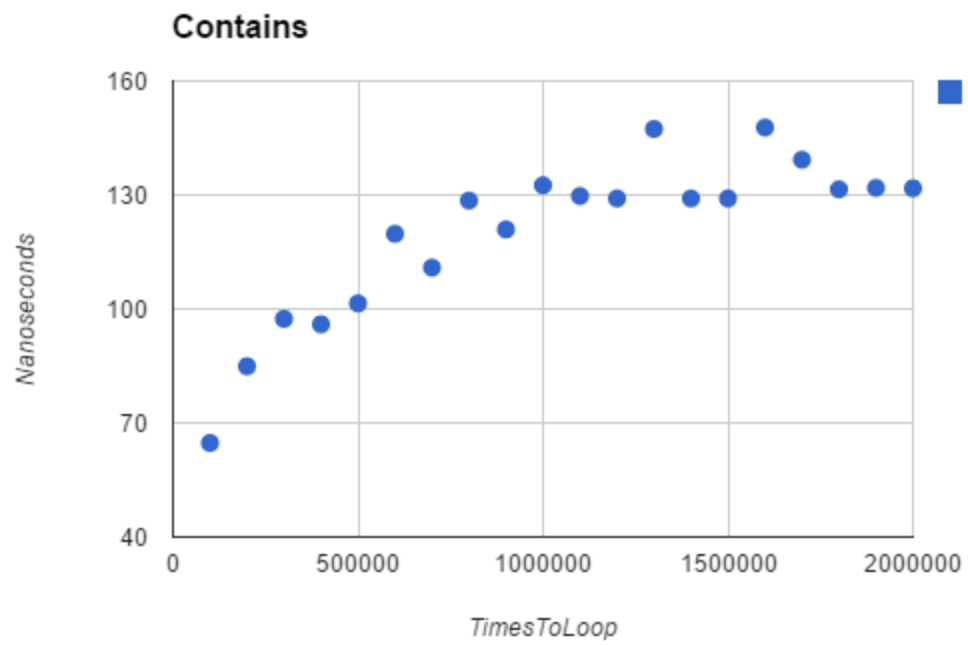1. Michael Swisher. I submitted the source code
2. Not very often/on timed increments. I was driver for about 75% of the time. No, I think better if I'm driving/hands on and he thinks better if he's navigating.
3. Michael was a great partner again this week. We ended up in the cade lab for about 10 hours on Thursday, and he never tried to say we should just give up and go home. Yes I will work with him again next week.
4. The add, contains, and remove methods would have already been implemented for us if we had used an arraylist. Development time might have been faster, but run time I am not sure about. Since we used binary search to loop, it had a complexity of log n; whereas arraylist iterates through the entire list from the start to finish to find the selected element. I believe our runtime is more efficient, but development time was less efficient.
5. I expect it to be O(log n) in the worst case since it uses binary search to look through the array.
6. Yes, it matches my prediction in question 5.

Add

7. Best case scenario is constant, worst-case scenario is O(log N);

Contains

Nanoseconds vs *TimesToLoop*

8. 20 hours