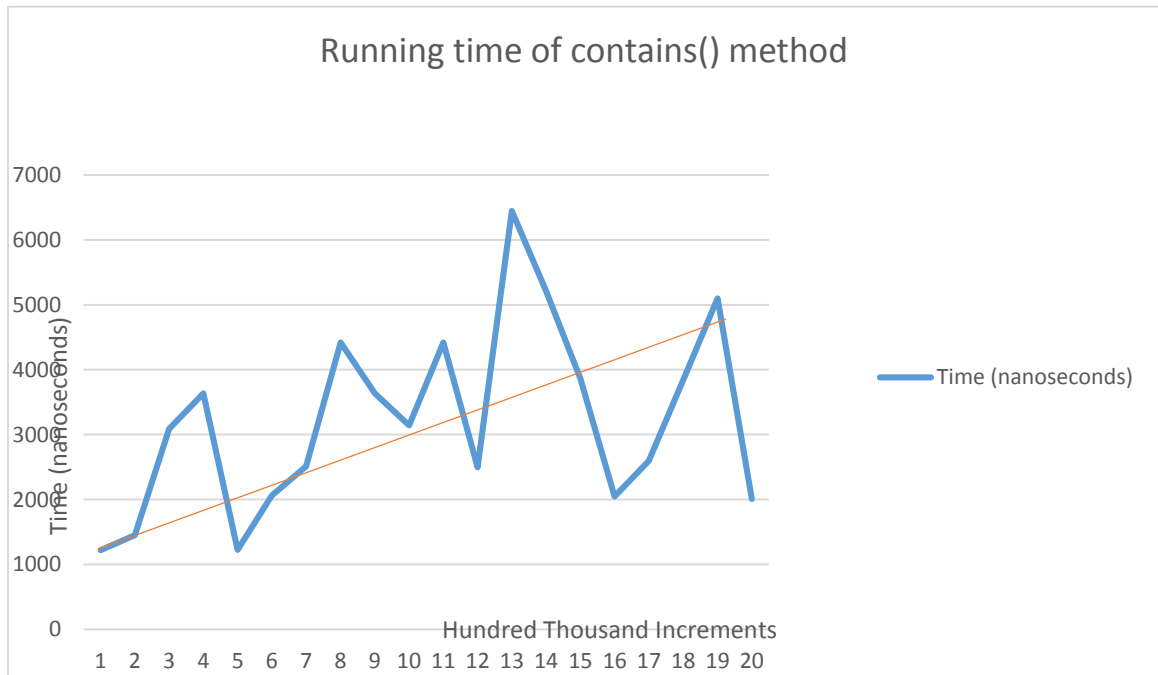


Analysis Document – Assignment 3

1. Josh Callahan is my partner. I'm submitting the code this week.
2. We switched on various intervals but generally between 15 and 30 minutes. It was a good rate of switching and whenever someone had an idea, they would take over and start coding. It usually just worked out really well that we took equal turns.
3. Josh is a great partner. We are almost on the exact same level of programming knowledge and where one lacks, the other has more experience and it works out really well. He works hard and is motivated. Yes I plan on working with him as much as possible.
4. Java's array list has methods that are already given and we wouldn't have to write so many of our own methods such as contains and add. We wouldn't have to keep track so much of the number of elements in the array since a Java List has a size() method that returns the size at any time.
5. Contains is pretty much just a binary search method which has  $O(\log n)$  behavior because it reduces the size of  $N$  by 2 every time it does a comparison.

6.



Set Size	Time (nanoseconds)
100000	1220
200000	1453
300000	3087
400000	3636
500000	1226
600000	2066
700000	2510
800000	4419
900000	3633
1000000	3142
1100000	4418

1200000	2495
1300000	6445
1400000	5212
1500000	3869
1600000	2047
1700000	2598
1800000	3845
1900000	5099
2000000	2012

The growth rate is approximately  $O(\log n)$  as it should be. The data is a little crazy but an approximate linear fit gives a growth that looks similar to a  $\log n$  growth.

7. The add method uses a binary search that is pretty much identical to the binary search used in the contains method. Therefore the timing will be very similar. It should have  $O(\log n)$  behavior. Worst case is  $O(\log n)$ .
8. We spent approximate 13 hours on this assignment.