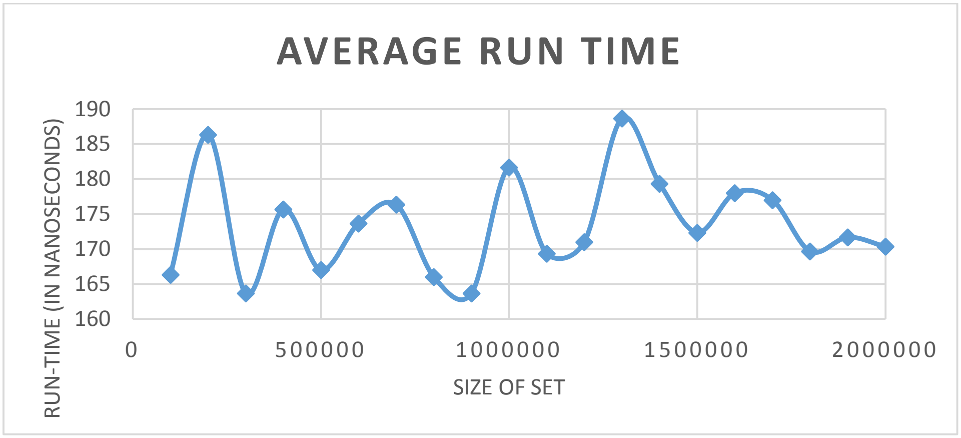
Tanner Barlow
CS 2420
February 4, 2015
Assignment 3 Analysis

1) Elijah Grubb is my programming partner. I submitted our source code.
2) We switched roles every 30-60 minutes. I would have liked to switch more, just because sometimes I felt like I had to fight for my idea if I wasn't the one typing at the time.
3) Elijah is great. He really knows his stuff. I definitely plan on working with him again and I feel like we're learning a lot together.
4) The only things that would have been different would be how the set grows. Lists already have the capability to add and remove items without having to do everything required with arrays backing it. With an array, we had to create a new array, copy everything over and then change the reference variable so that it points to the new array.
5) Because of the halving principle (contains uses a binary search-like method) the complexity of the contains method would be O(log n).
6) TimesToLoop: 100000

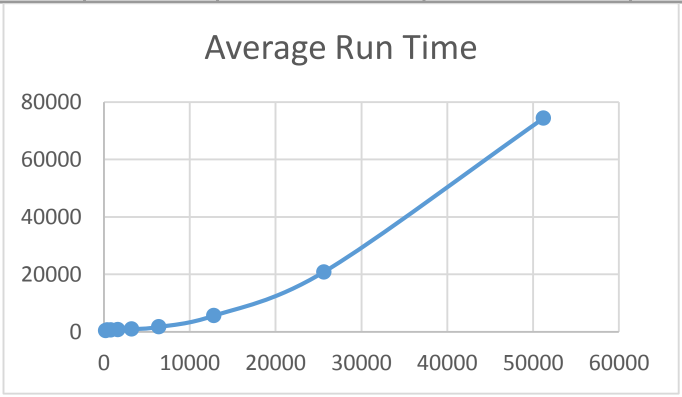
	Run Time 1	Run Time 2	Run Time 3	Average Run Time
100000	155	196	148	166.3333333
200000	189	181	189	186.3333333
300000	159	177	155	163.6666667
400000	178	173	176	175.6666667
500000	174	158	169	167
600000	168	185	168	173.6666667
700000	170	179	180	176.3333333
800000	164	163	171	166
900000	163	168	160	163.6666667
1000000	174	211	160	181.6666667
1100000	165	170	173	169.3333333
1200000	160	185	168	171
1300000	177	218	171	188.6666667
1400000	189	163	186	179.3333333
1500000	167	169	181	172.3333333
1600000	198	166	170	178
1700000	188	181	162	177
1800000	177	164	168	169.6666667
1900000	178	172	165	171.6666667
2000000	170	171	170	170.3333333

Tanner Barlow
CS 2420
February 4, 2015
Assignment 3 Analysis



7) Below is my plot of running times. In the worst case, it seems to be following a logarithmic pattern multiplied by N, which makes sense because a binary search is logarithmic and to insert the items would be N units of work. O(N log(N))

Set Size	Times to Loop	Run time 1 (nanoseconds)	Run time 2 (nanoseconds)	Run time 3 (nanoseconds)	Average Run Time
200	1000000	314	538	307	**386.3333**
400	1000000	720	476	818	**671.3333**
800	1000000	638	681	597	**638.6667**
1600	1000000	625	766	743	**711.3333**
3200	1000000	971	914	873	**919.3333**
6400	1000000	1687	1692	1747	**1708.667**
12800	1000000	5560	5633	5658	**5617**
25600	1000000	21309	20639	20541	**20829.67**
51200	1000000	74803	74172	74213	**74396**



8) I spent, including the analysis paper, a total of 8-9 hours on this assignment