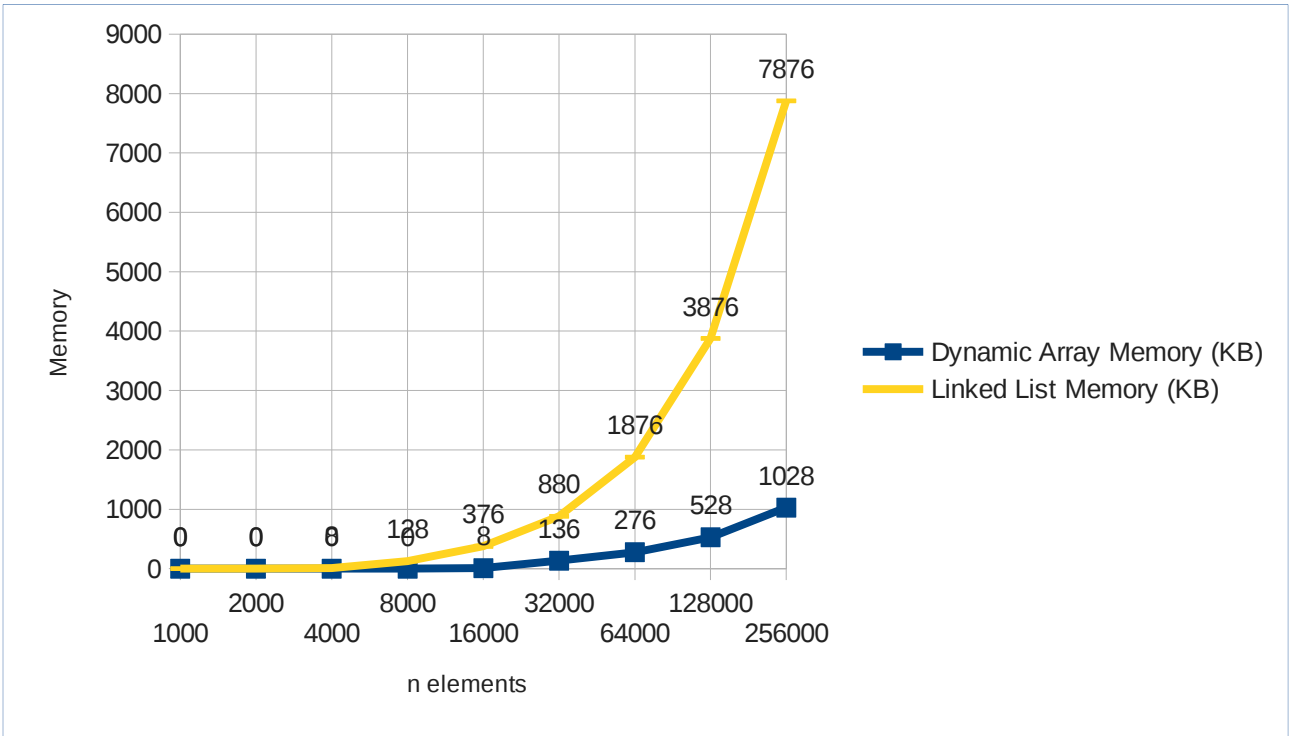
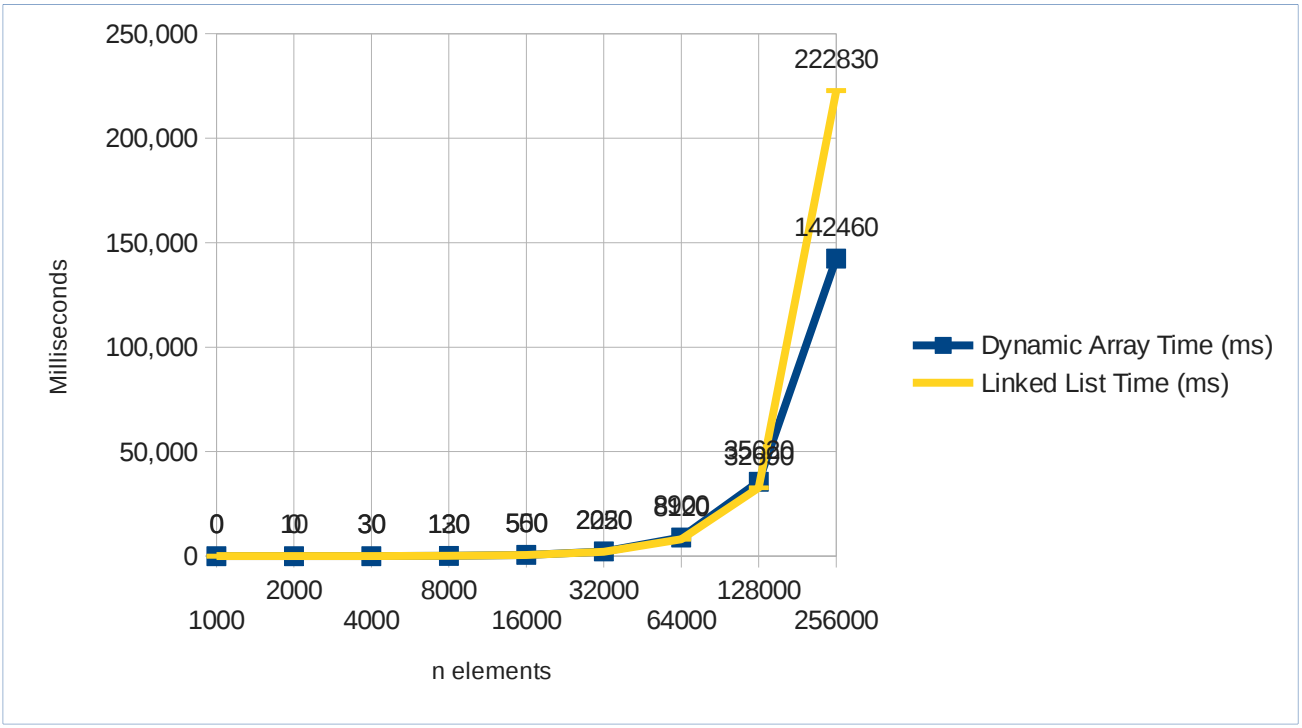
Brian Wilkins
CS261-400 Fall 2015
Assignment 3 Part 2
10/25/2015

n	Dynamic Array Memory (KB)	Dynamic Array Time (ms)
1000	0	0
2000	0	10
4000	0	30
8000	0	130
16000	8	550
32000	136	2,220
64000	276	8,900
128000	528	35,620
256000	1028	142,460

n	Linked List Memory (KB)	Linked List Time (ms)
1000	0	0
2000	0	0
4000	8	30
8000	128	120
16000	376	500
32000	880	2,050
64000	1876	8,120
128000	3876	32,690
256000	7876	222,830


9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Memory
7876
3876
1876
880
376
128
1028
528
276
136
8
0
0
0
0
1000
2000
4000
8000
16000
32000
64000
128000
256000
n elements
Dynamic Array Memory (KB)
Linked List Memory (KB)


250,000
200,000
150,000
100,000
50,000
0
Milliseconds
222830
142460
0
1000
2000
4000
8000
16000
32000
64000
128000
256000
n elements
Dynamic Array Time (ms)
Linked List Time (ms)

Questions:
1. Which of the implementations uses more memory? Explain why.
The linked list uses more memory because it needs to allocate additional memory to the head and tail pointers, whereas the dynamic array only has a single sentinel pointer.

2. Which of the implementations is the fastest? Explain why.
This depends on the amount of elements. Linked lists are fastest up to 128,000 items. However, when testing for 256,000 items, the dynamic array is fastest. This could be due to it being easier to find indexed array items over items in a linked list, even though there is a sentinel.

3. Would you expect anything to change if the loop performed remove() instead of contains()? If so, what?
For the linked list I do not expect there to be a difference in performance because the functions are both O(n).