- Which of the implementations uses more memory? Explain why.

  The Linked List Implementation uses more memory. This is because the Linked List must store the value, next and previous locations and the Dynamic Array only needs to store the value.

- Which of the implementations is the fastest? Explain why.

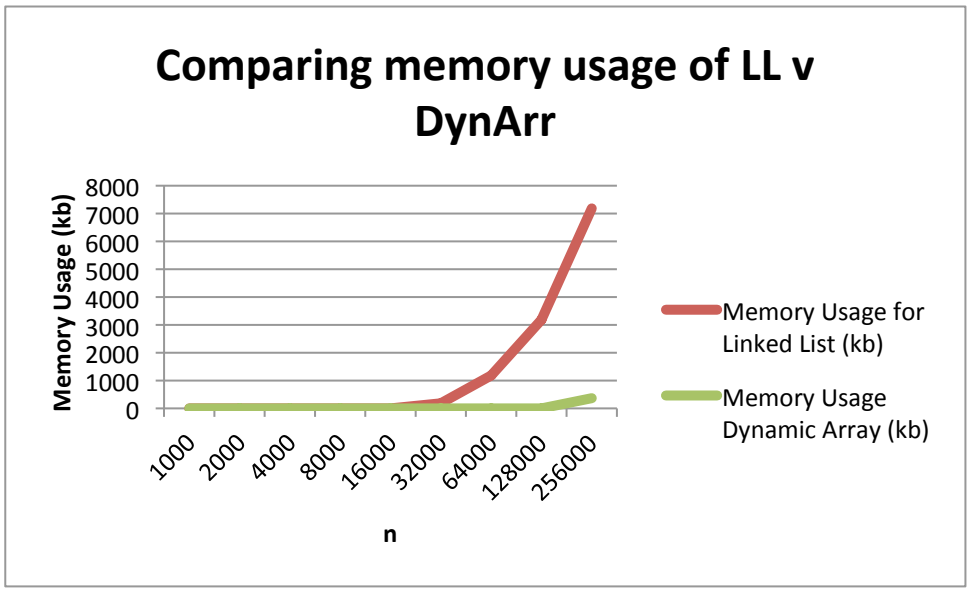
  The Dynamic Array Implementation runs contains() the fastest. This is because searching a Dynamic array is O(1) run time, while searching a Linked List is O(n).

- Would you expect anything to change if the loop performed remove() instead of contains()? If so, what?

  I would expect that the Linked List would be slightly faster than the Dynamic Array, because suddenly once the Dynamic Array finds the proper element to remove it must now update all the other elements from that index up to n, slowing its performance drastically to O(n) at worst case. Linked List only needs to remove the connections and deallocate the memory which is O(1) run time added to the already O(n). Nevertheless, the two run times would become O(n), with Linked Lists coming out just a smidge ahead if you really measured down to an exact run time calculation.

Memory Usage LL v DynArr

n	Memory Usage for Linked List (kb)	Memory Usage Dynamic Array (kb)
1000	0	0
2000	0	0
4000	0	0
8000	0	0
16000	0	0
32000	180	0
64000	1184	0
128000	3184	0
256000	7184	368



Time for contains() LL v Dyn Arr

n	Time for Linked List (ms)	Time for Dynamic Array (ms)
1000	0	0
2000	0	10
4000	30	30
8000	130	150
16000	530	610
32000	2140	2440
64000	8570	9780
128000	34410	39150
256000	226340	156600