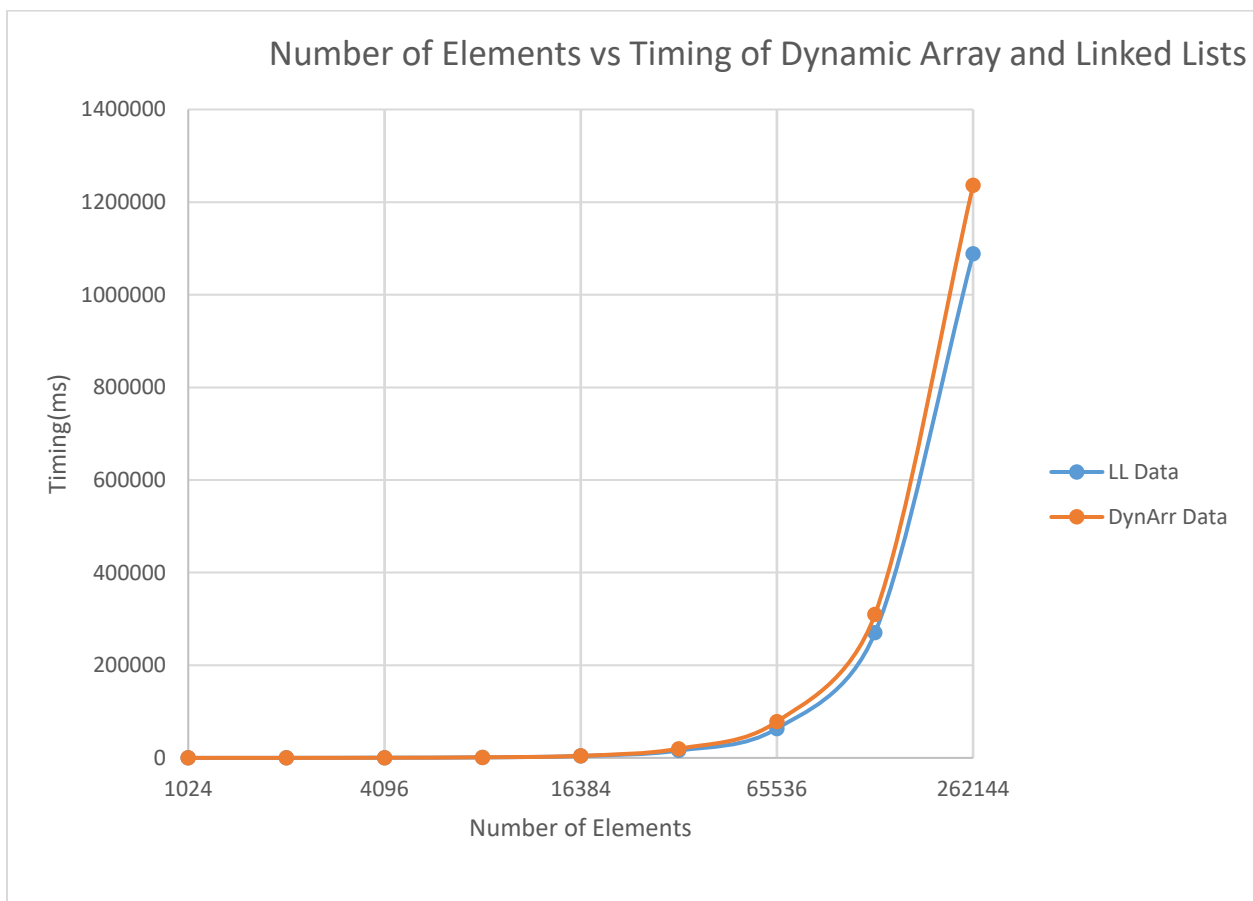
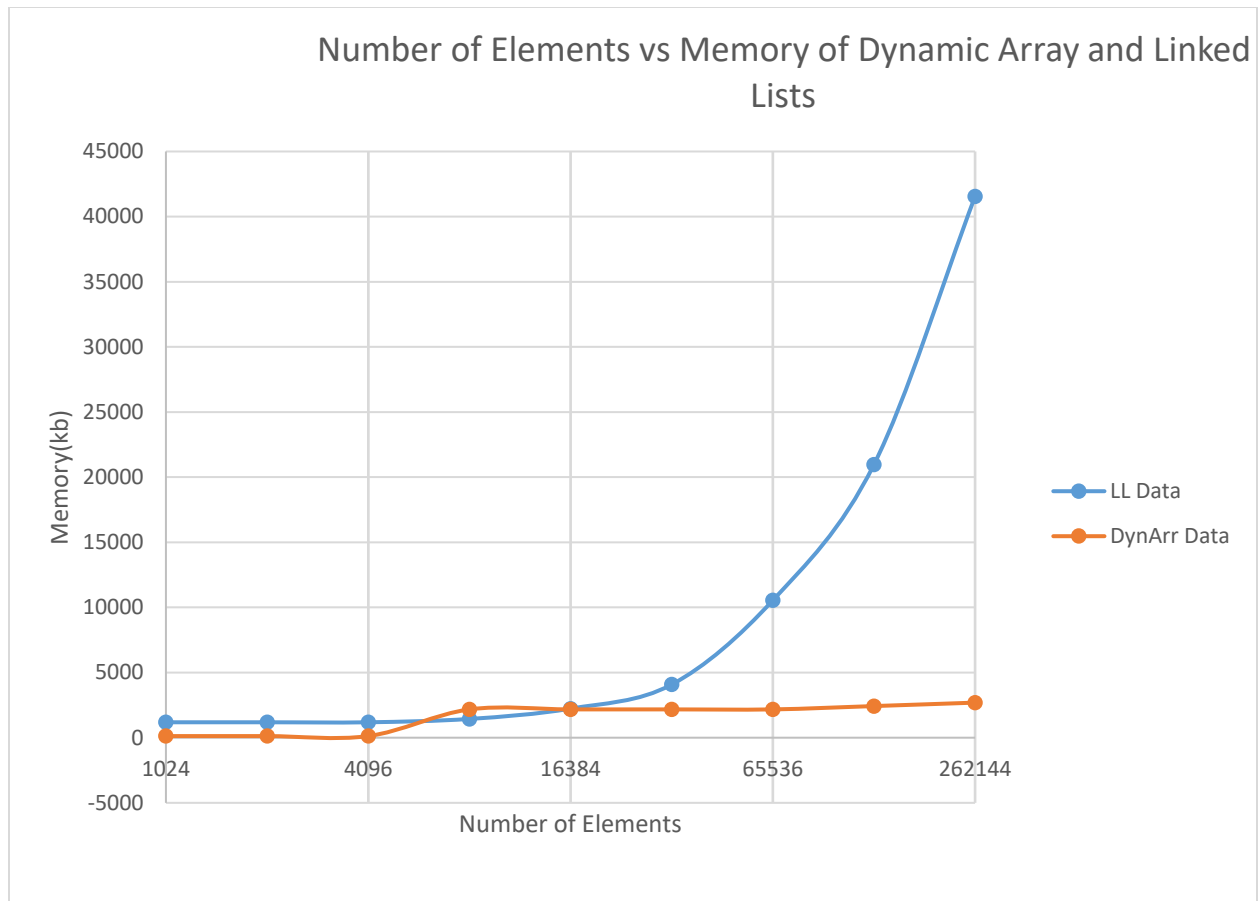


Brandon Lo

	LL mem	DynArr mem	LL timing	DynArr timing
2^10	1184 KB	124 KB	20 ms	20 ms
2^11	1184	124	70	80
2^12	1184	124	450	320
2^13	1440	2172	1010	1250
2^14	2232	2172	4130	4490
2^15	4080	2172	15860	19850
2^16	10544	2172	63420	78410
2^17	20968	2428	270260	309870
2^18	41552	2692	1088820	1236130





Which of the implementations uses more memory? Explain why.

The Linked List uses more memory. This is because each link contains the value, as well as pointers to the adjacent links. Therefore, there is more memory that has to be stored per value.

Which of the implementations is the fastest? Explain why.

The Linked List is the fastest because when a link is made it does need to copy the entire list over instead a sentinel is used to help add the link in. On the other hand a dynamic array needs to copy the entire array to the new memory area when the array is resized. This is proven by the plot.

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

No I would not expect anything to change between the loops since they are both $O(n)$ operations which iterate through the loop and look for a value.