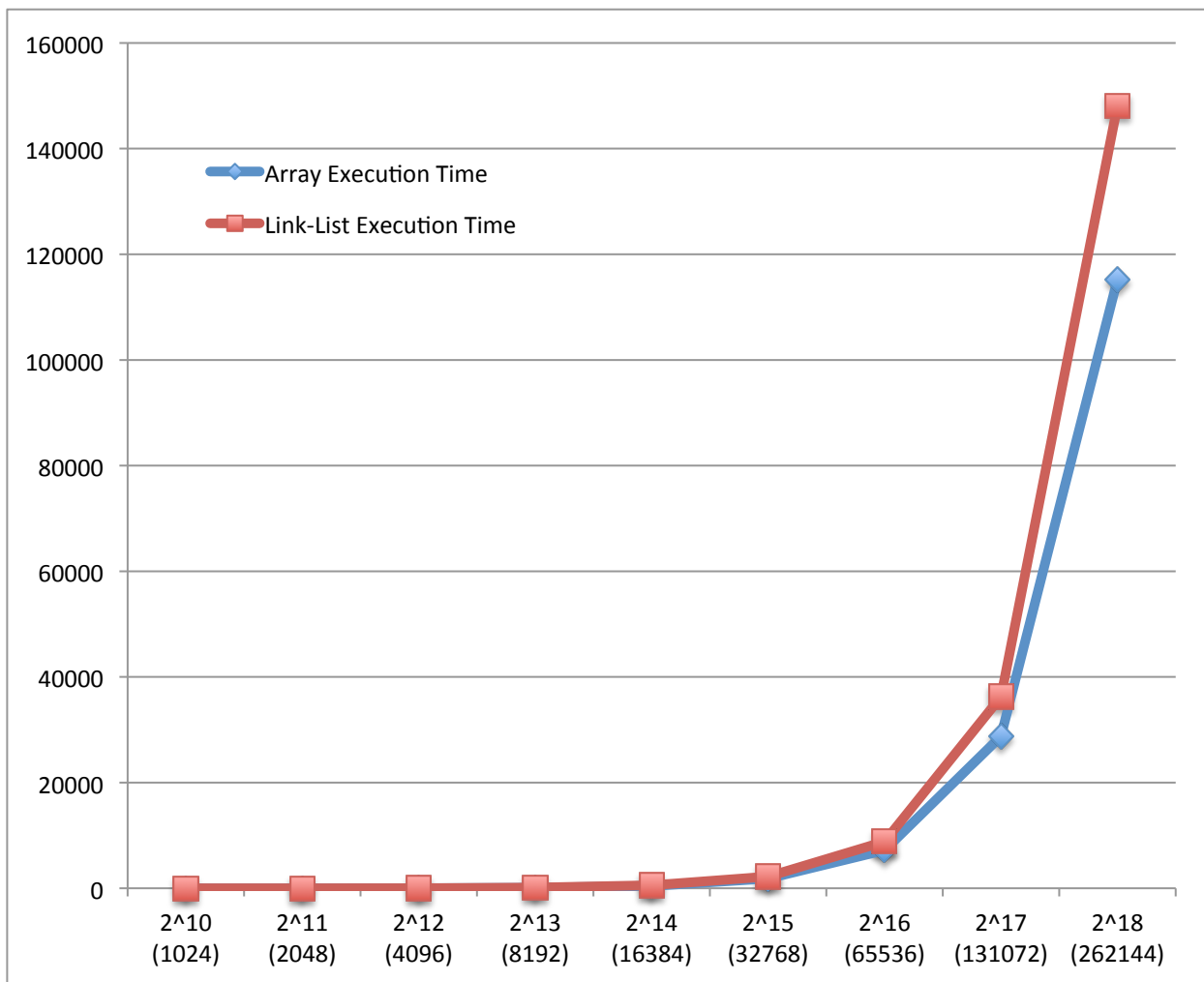
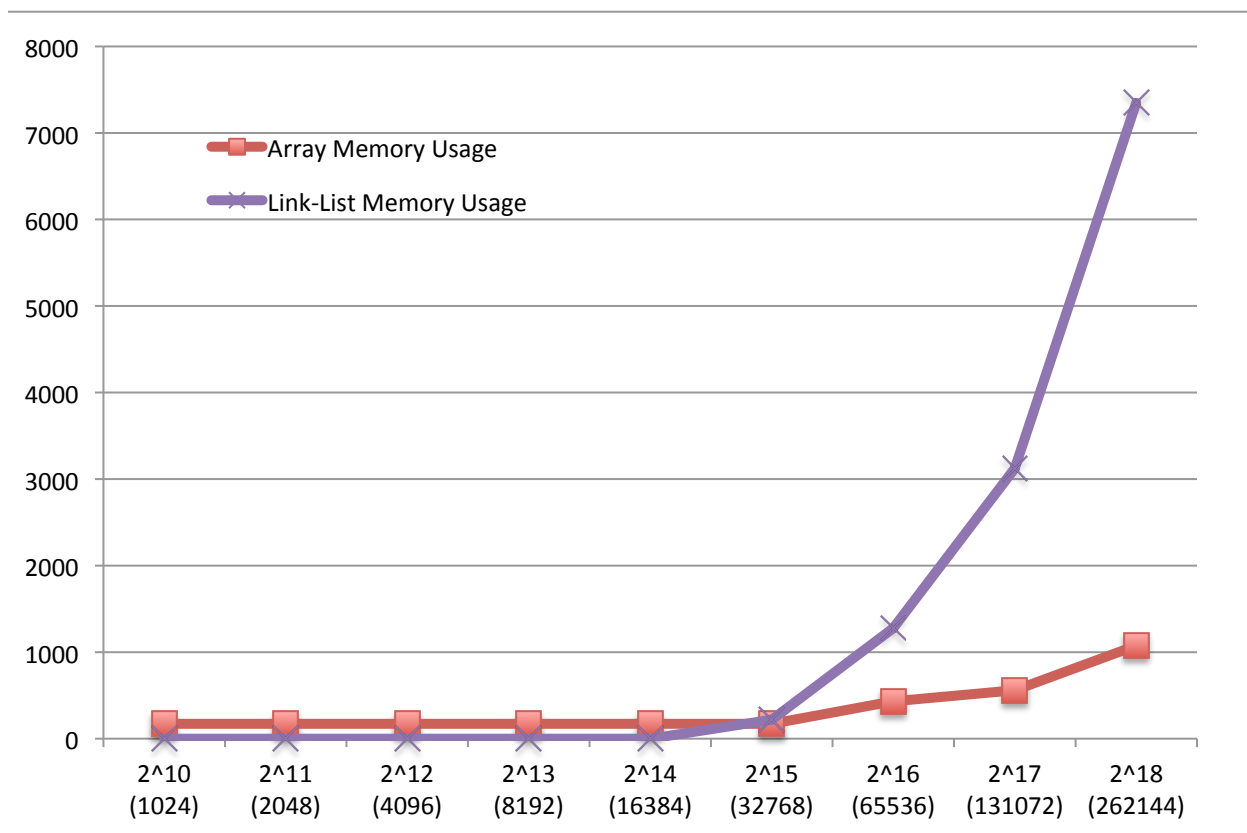


Linked-List vs Dynamic Array Bag Implementation Comparison

Elements	Dynamic Array Bag		Linked List Bag	
	Time (ms)	Memory(KB)	Time (ms)	Memory(KB)
2^10 (1024)	0	172	0	0
2^11 (2048)	0	172	0	0
2^12 (4096)	20	172	30	0
2^13 (8192)	110	172	130	0
2^14 (16384)	440	172	560	0
2^15 (32768)	1820	172	2200	220
2^16 (65536)	7180	432	8840	1276
2^17 (131072)	28780	560	36270	3124
2^18 (262144)	115230	1076	148120	7348





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

The linked-list uses the most memory as n approaches infinity because each link requires more data than a single array element, so although the linked-list is more efficient to start, once n grows very large it consumes a lot more memory resources than the dynamic array.

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

The dynamic array is faster because the data is stored contiguously, whereas the linked-list has to follow its pointers from one random memory location to another, thus taking more time to perform the operation.

3. Would you expect anything to change if the loop performed `remove()` instead of `contains()`?

If so, why? Yes, I would expect the dynamic array deque structure would be even slower because not only will it search for an element, but it would then also have to shift all the subsequent elements with a larger index value ($O(n)$). The linked-list wouldn't experience a measurable difference.