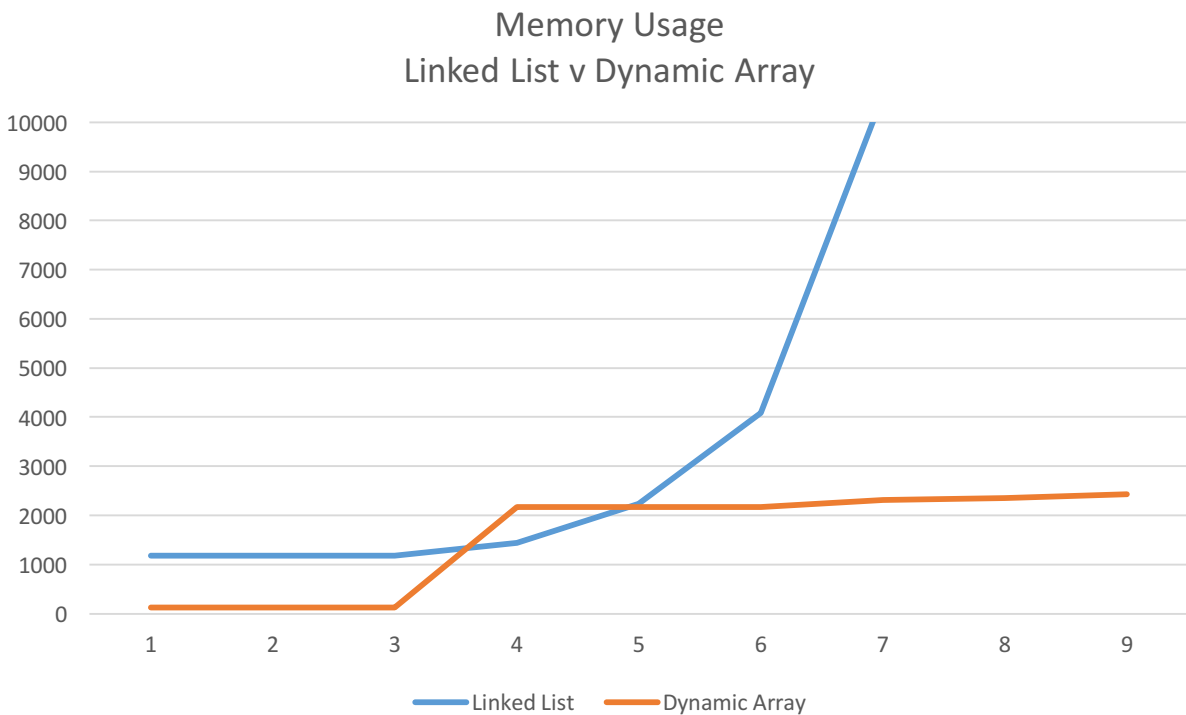


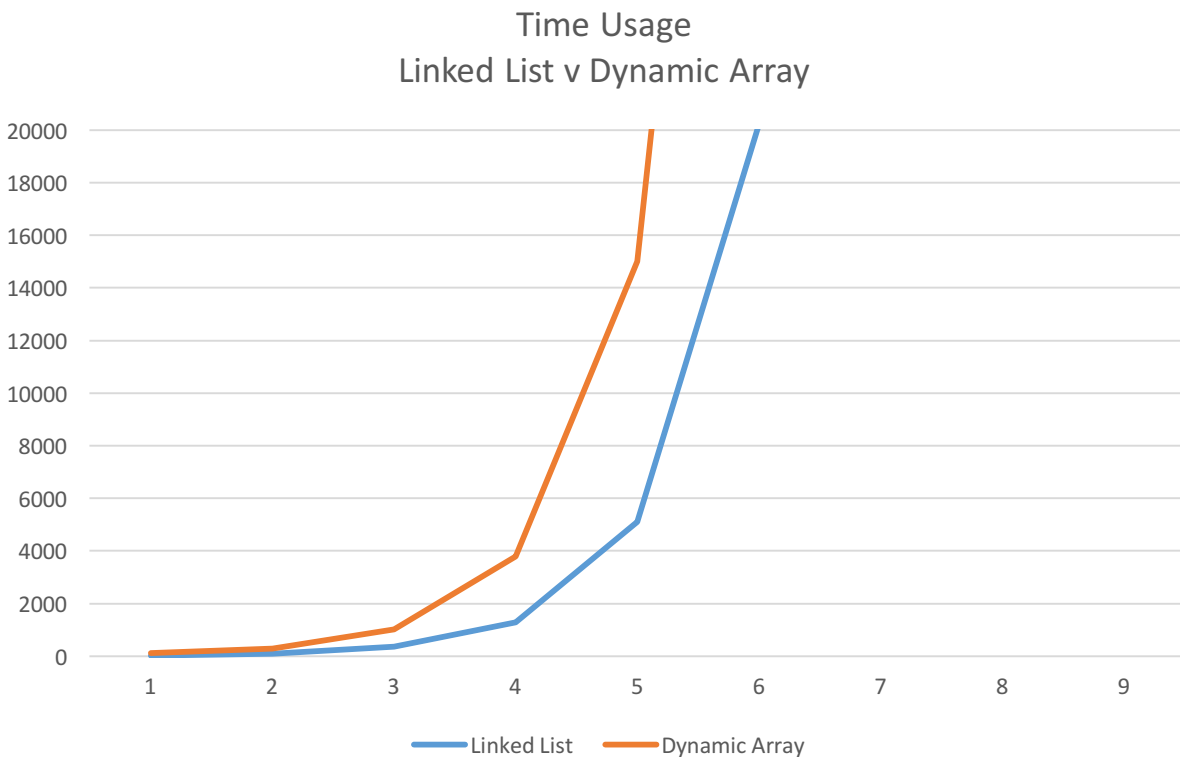
Student: Joaquin Saldana

Assignment 3 / Problem 2

Below are my findings from the test between the dynamic array and the linked list implementations:

| Assignment 3 Plot Chart | | | | | |
|-------------------------|------------|-------------------|---------------|---------------------|---------------|
| | | Memory Usage (KB) | | Time (Milliseconds) | |
| 2^n | n elements | Linked List | Dynamic Array | Linked List | Dynamic Array |
| 10 | 1024 | 1184 | 124 | 20 | 120 |
| 11 | 2048 | 1184 | 124 | 90 | 280 |
| 12 | 4096 | 1184 | 124 | 370 | 1020 |
| 13 | 8192 | 1440 | 2172 | 1290 | 3800 |
| 14 | 16384 | 2232 | 2172 | 5100 | 15010 |
| 15 | 32768 | 4080 | 2172 | 20280 | 57920 |
| 16 | 65536 | 10544 | 2308 | 85640 | 231780 |
| 17 | 131072 | 20968 | 2356 | 325630 | >600000 |
| 18 | 262144 | 41552 | 2436 | >60000 | >600000 |





Below are my answers to the questions presented in problem 2 of the assignment:

Which of the implementations uses more memory? Explain why:

In the exams performed, it's relatively obvious the Linked List implementation absorbed more memory than the dynamic array implementation. The linked list implementation has a lot of memory overhead in which it needs to not only allocate memory for the list, but for the links, the sentinals, and the front and previous pointers. This in turn creates a great stress on the memory needed to perform the necessary operations.

Which of the implementation is the fastest? Explain why:

The dynamic array was the fastest. Since array is in a continuous chunk of memory, the function calls do not need to perform any large calculations for offsets of addresses between various pointers. This results in faster traversals of the structure and better performance.

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

With remove, for linked list and dynamic array, the `remove()` function is an $O(1)$ operation, at worst, each time no matter what. Meanwhile, a `contains()` function at worst can be an $O(n)$ operation. And the reason for the $O(n)$ operations, is because you will need to traverse the entire lists, linked list or dynamic array, in order to find the element currently looking for.