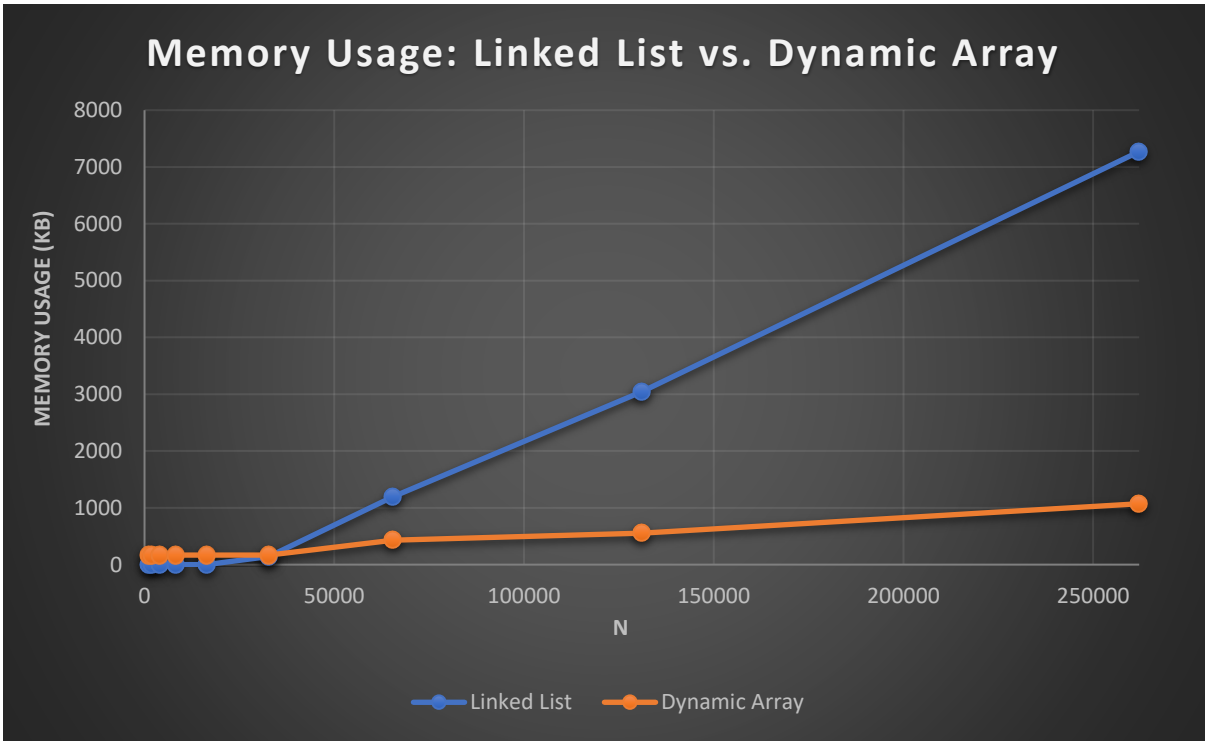
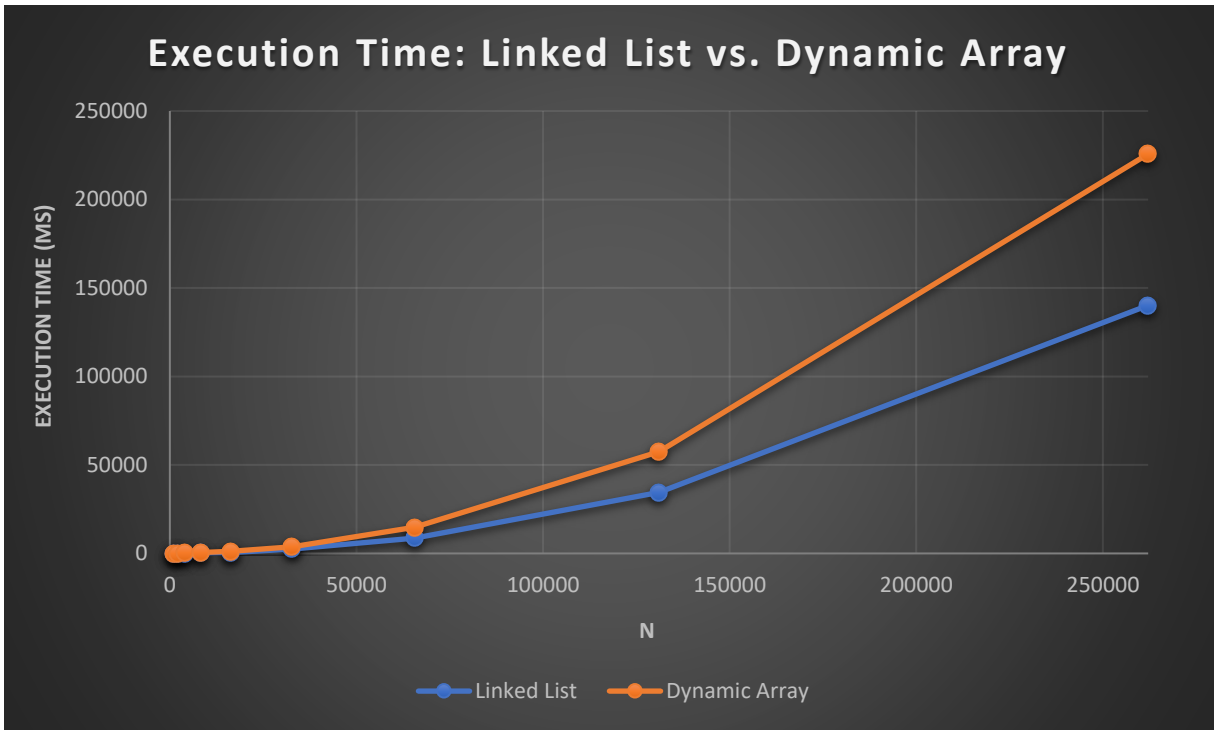
Christopher Merrill
CS261
1/31/2017

Programming Assignment 3 Part 2: Performance Comparison





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

The linked list implementation uses more memory because the memory required to store each link structure is greater than the memory required to store an integer in the dynamic array. As n increases, this slight difference in size begins to show on a large scale (as seen in the above graph).

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

The link list is implementation is the fastest. This is due to the fact that the cost to add a new link is constant (for each new link all that needs to happen is a few simple pointer updates). The dynamic array, on the other hand, has to increase capacity every time it is filled, which adds time that isn't necessary with the linked list. In other words the cost to add an item in the linked list is O(1) whereas the cost to add an item into the dynamic array is O(1)+,

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

I would expect the dynamic array implementation to take even longer if the loop performed remove() because, in order to remove an item from the dynamic array, all items after that item must be shifted over. This would require significantly more operations than the contain() function, since contain() doesn't require a shift.

The execution time of the linked list implementation shouldn't be affected nearly as much as the dynamic array, but it may take slightly longer as it requires a few pointer updates with remove() that aren't necessary with contain().