

Mathew McDade

10/15/2017 1:57:05 PM

Assignment 3 - Problem 2: Linked List vs Dynamic Array performance

Note: All graphs can be found on page 2.

1. Which of the implementations uses more memory?

According to program outputs, and as can be seen on the graphs, starting around $N=2^{14}$ the linked list implementation uses significantly more memory than the dynamic array. This trend continues and at $N=2^{17}$, the linked list uses about five times more memory than the dynamic array. This is the expected result considering that each value stored in the dynamic array only requires the space for the given value, while each additional value in the linked list requires space for the value, as well as a pointer to the next and previous links.

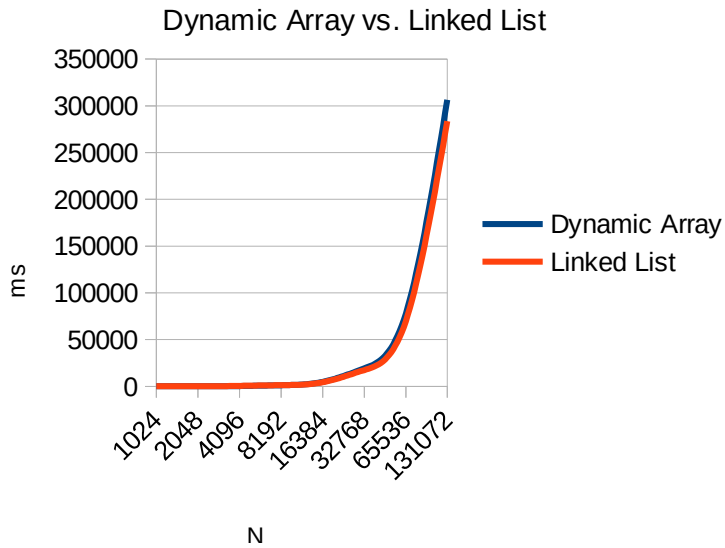
2. Which of the implementations is the fastest?

When run with valgrind, the two implementations ran contains() at very similar speeds for all values of N. Without valgrind, the dynamic array ran slightly faster for all values of N. Interestingly, both implementations took about ten times longer to run with valgrind as opposed to without for all values of N. This is a somewhat different result than I expected. My expectation was that the dynamic array would run faster because it is iterating over a continuous block of memory, while the links of the linked list may be stored non-contiguously. I'm unsure whether this is due to an excellent linked list implementation or a poor dynamic array implementation. The timing results run without valgrind, averaged over multiple test runs, are more reflective of the expected result. Additional profiling using the time and gprof tools reflected similar results to the no valgrind results.

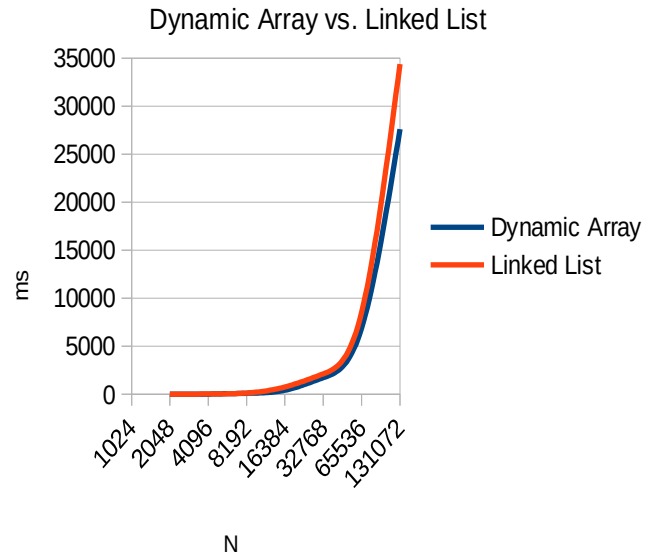
3. Would you expect anything to change if the loop performed remove() instead of contains?

Yes, assuming the removes are occurring somewhere besides the very end of the data structures, I would expect the dynamic array to perform this task more slowly than the linked list due to the need to shift all values at indexes greater than the removed element to be shifted left each time, whereas the linked list need only update the pointers before and after the removed item. I constructed this test case, and the results seen in the graph do reflect that the dynamic array will take an increasingly larger amount of time to run removes() when compared to the linked list.

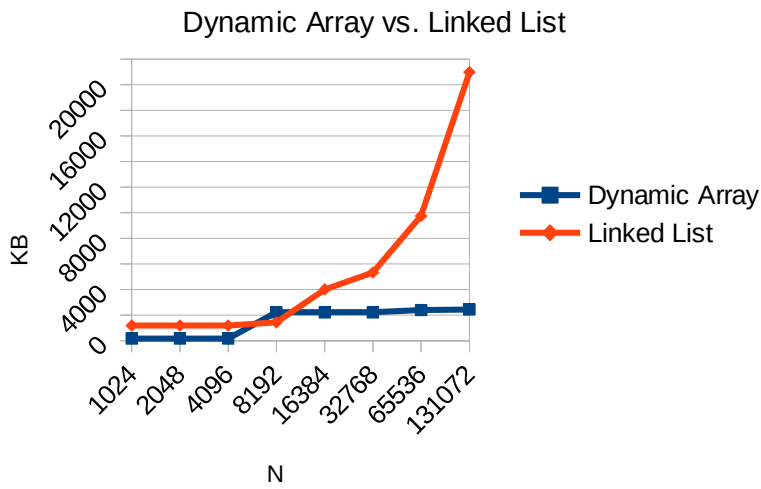
Time for Running Contains()



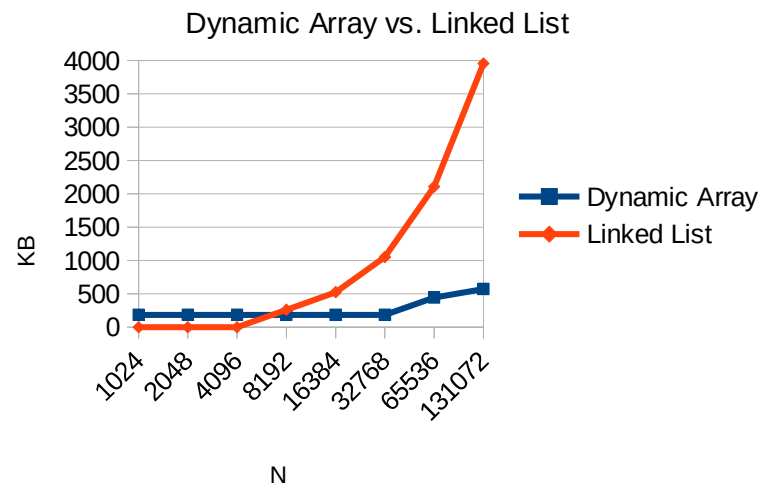
Time for Running Contains() - No Valgrind



Memory Used by



Memory Used by - No Valgrind



Time for Running Remove()

