

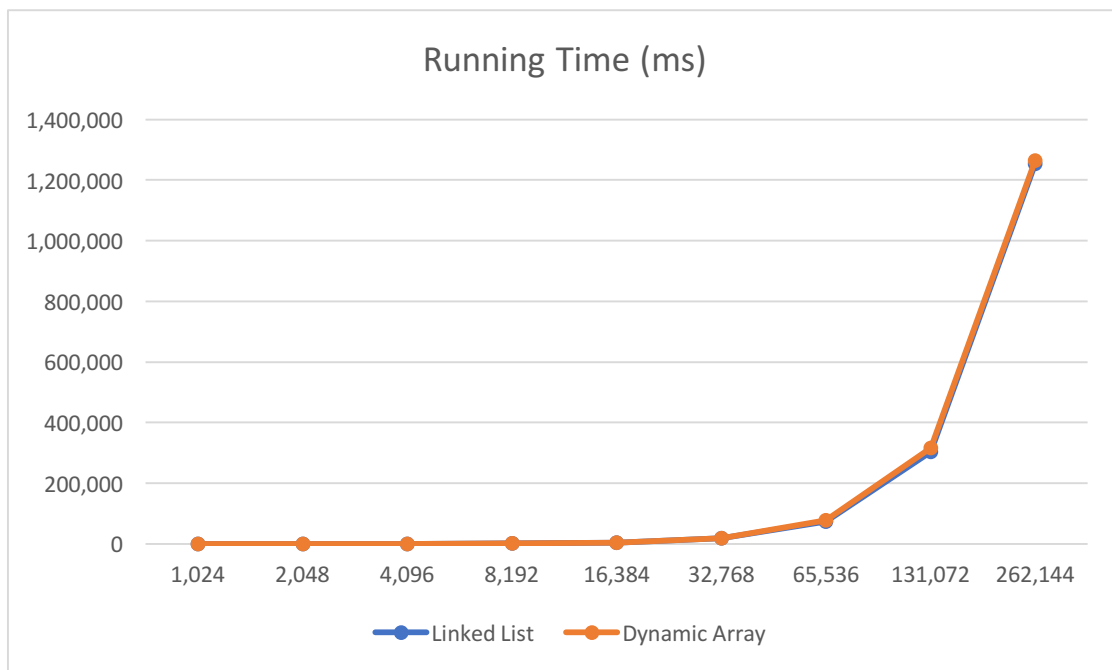
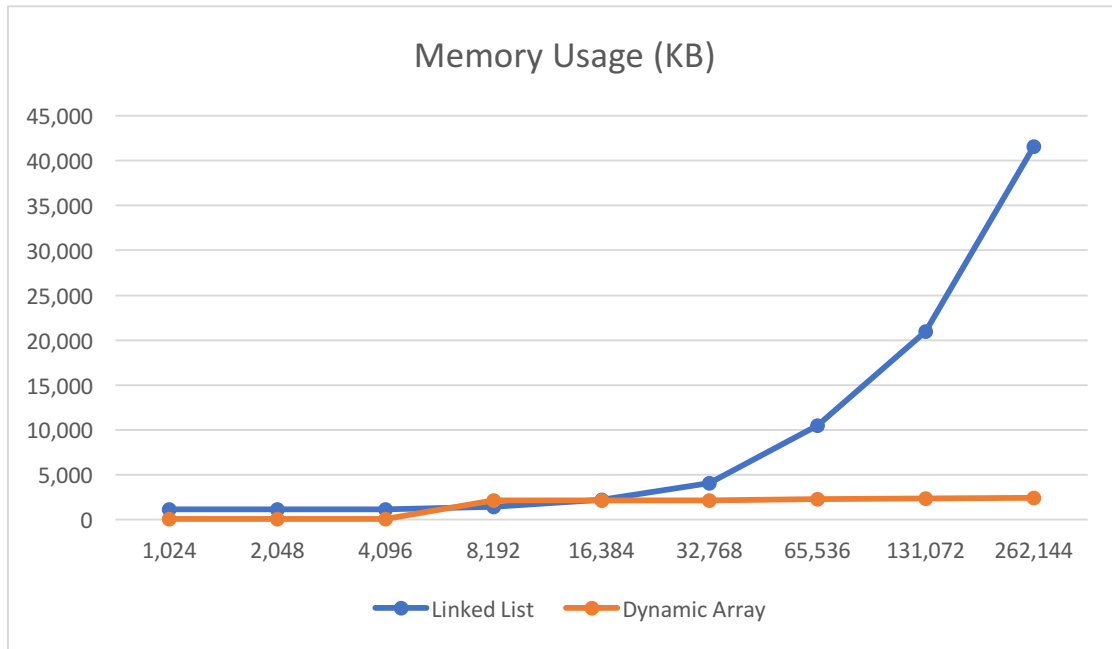
CS 261 Data Structures

Pedro Torres Picón

Assignment 3 -- Linked List Variations

Problem 2: Linked List vs Dynamic Array performance comparison

The following two plots were created with the results of running `linkedListMain.c` and `dynamicArrayMain.c` using Valgrind on the flip server. They show the difference in memory and time performance for each implementation.



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

Except for a brief crossover at 2^{13} , the Linked List uses more memory when the program is run with Valgrind. As the input size grows, the memory used by the Linked List grows **much** faster than the memory used by the Dynamic Array. This is because with a Dynamic Array we only need to store the values and a pointer to the first item, while with a Linked List we have to store the value, plus two pointers (next, prev) for *each* node in the list.

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

They run in about the same time, from the beginning and going up to large inputs. This is because in both implementations we are traversing the array by using a memory address to access the next element until we check every element for the value we are looking for. The array implementation takes *slightly* longer, and I assume it might be because of the small time it takes to compute the address of an array element from the first array plus its offset.

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

No. This is because `remove()` is also a traversal operation, and the fact that we are looking to remove an item or find an item makes no real difference. The Linked List and the Array will perform similarly.