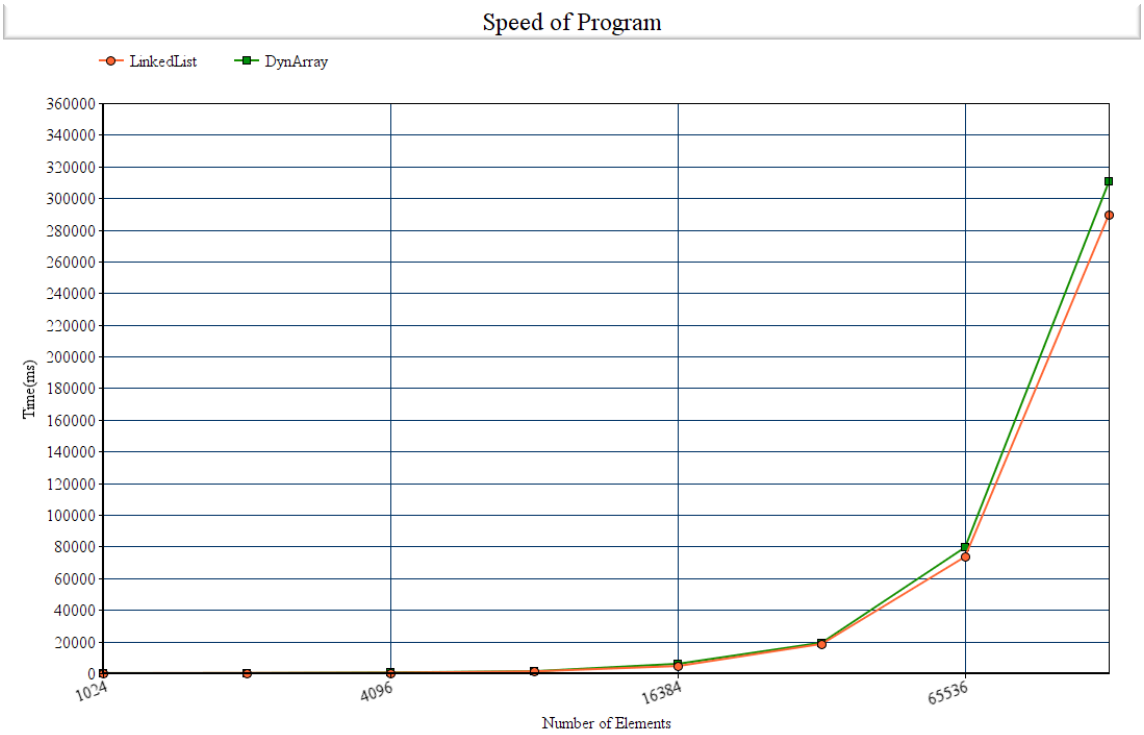
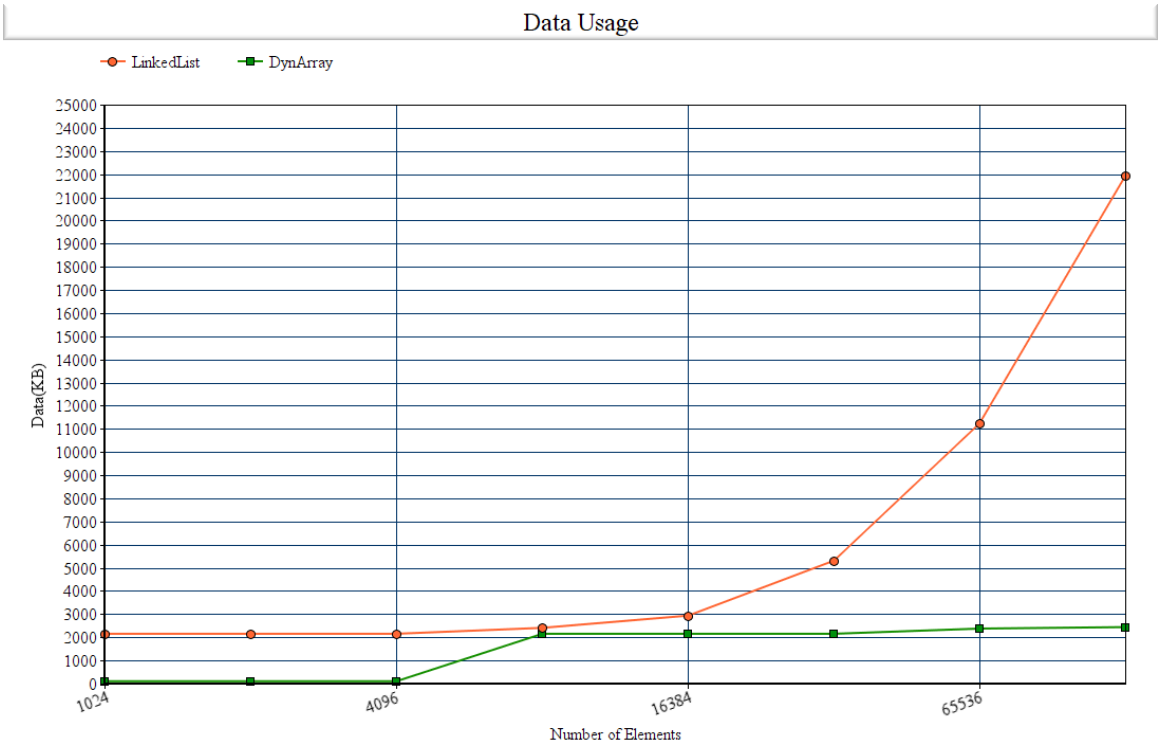
Ethan Dunham
Assignment 3 part 2
04/28/17





1. Which of the implementations uses more memory? Explain why.
   The linked list appears to use more memory. I would say that the reason for this is that it has to create a whole struct for each value instead of just sticking it in an array. I'm not sure that the memory check worked correctly on the dynArray, but those are the files that were provided and were run through Valgrind. I expected the array's memory usage to double each run due to the array having to double in size 1 extra time every consecutive run, but I guess I was incorrect.

2. Which of the implementations is the fastest? Explain why.
   DynArray is a little faster due to more aspects being O(1) than the linked list. I would have expected the DynArray to be slower due to it having O(1)+ add instead of O(1) of linked list. Perhaps the act of creating the struct for the linked list makes it slower.

3. Would you expect anything to change if the loop performed remove() instead of contains()? If so, why?
   I think they would remain the same because the remove is O(N) for both a linked list and dynamic array.