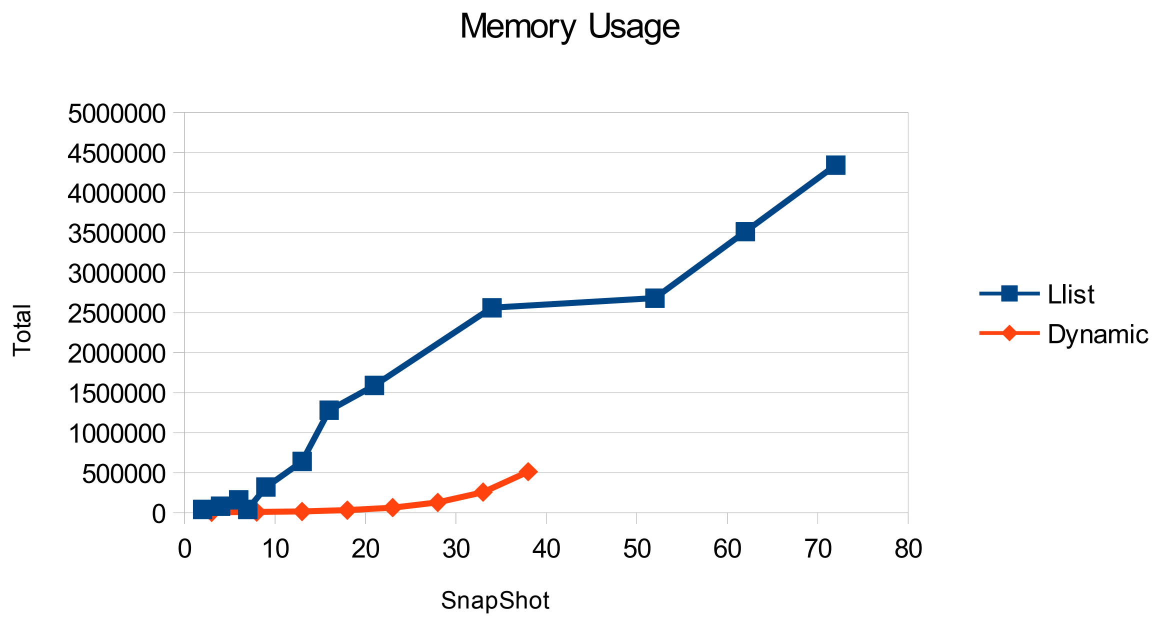
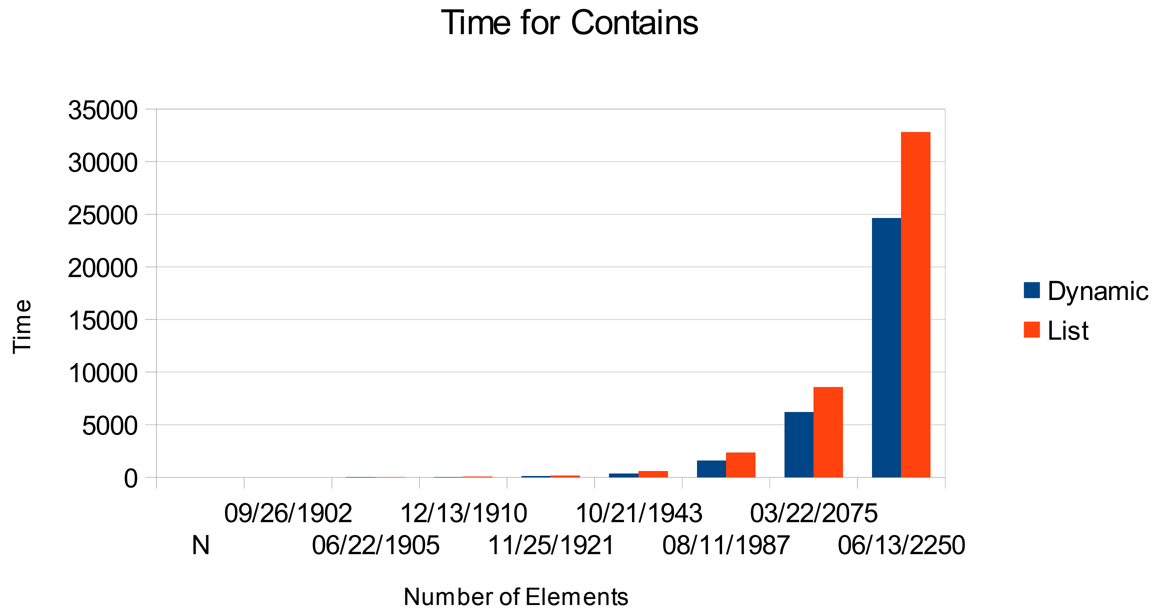
Alex Wilson and Braden Ackles





1.) The linked lists use more memory. When allocating memory the Dynamic Arrays only need to allocate the space needed to store values while the Linked List needs to allocate enough memory for a new link struct. A link struct takes up the memory to hold the value and the space needed for pointers while the array is dedicated just to values.

2.) The Dynamic Array runs faster than the Linked List. This is because the Dynamic Array can directly access an element stored within in while the Linked List has to follow pointers before it can check what value is stored in each individual struct. So the Dynamic Array just has to increment its iterator by 1 to check the next element while the Linked List has to change a pointer. It is also faster to allocate one big chunk of memory for the Dynamic Array than to allocate each link in a Linked List

3.) The Time would remain the same but the Memory usage would change. Unless you resized the Dynamic Array when it got too small the Dynamic Array could end up taking up a larger amount of memory. Unlike the Linked List all memory for the dynamic array happens at once and it normally doesn't shrink as the amount of data stored in it is removed. The Linked List on the other hand free unused links leading so there is not wasted space.