

Andrew Ortega  
Linked List Project  
3530

What is the computational complexity of the methods in the implementation?

There are seven commands which allow users to interact with the linked list. These functions are implemented in 5 methods with respective task. A variable called nodecounter keeps an active count of the number of lines in the list to ensure out of bound nodes cannot be accessed.

I combined the InsertEnd and Insert method. If the node is being inserted at the end, it checks to see if head is null in  $O(1)$  time. If head is not null, the list must be traversed until the last node is found and then the new node is linked to the end. This operation takes  $O(n)$  time. If the node is being inserted into a position,

Delete and edit and take  $O(n)$  time, they use a for loop to traverse the list  $n$  times to the desired position to be changed, removed or printed.

Search is also  $O(n)$ . It traverses the list until it reaches the end looking for a string of data. I used the C++ .find function to search each node of the list, I am unsure of the behind the scenes working of this but some of the documentation I was reading said it was linear.

Your thoughts on the use of linked lists for implementing a line editor. What are the advantages and disadvantages?

The linked list line editor had  $O(n)$  complexity in most cases which allows it to be pretty efficient. I only implemented a singly linked list and one of the disadvantages is if you are doing multiple edits to consecutive lines the list needs to be traversed each time.

What did you learn from this assignment and what would you do differently if you had to start over?

I learned a lot about segmentation faults throughout my debugging process. Until my nodecounter and other various checks were implemented my program crashed numerous times from me trying to access an area of memory which was out of bounds of the program. Not clearing the input buffer was also caused segmentation faults. There were a few C++ functions which came in handy such as stoi and .find which I haven't used before.