Madeline Farina
COP3530, FALL2018
Programming Assignment 1 reflection

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

Because all of my methods involved a standard linked list with no recursion or nested loops, my insert and delete methods were O(1) and all the other methods were O(n).

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

In general, a linked list has its advantages and disadvantages in that it is useful when you don't know how many elements you are going to store at initialization, allowing for a seemingly infinite amount of user input, and instead of storing data in one spot of memory, it can be stored in multiple places. A node of a linked list can also contain as much data as you want as long you set it up correctly, so for a line editor you could have nodes that also contain data like size or a boolean isFree. It does have its disadvantages though when compared to array-based lists, like array-based lists are better when insertions are mostly at the end of the list and when access is the most common operation. Also, array-based implementation requires storage of only the item while linked lists require storage of the item and pointer, so technically a linked list implementation requires more memory. For time for access, array implementation allows O(1) constant time access to any index while linked list implementation requires stepping through the list to access an item; consequently, runtime is O(n) and thus slower than an array implementation.

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

I began this assignment with the expectation that it would be something fairly straight-forward and easy because we had already done linked list exercises, and I've had some previous experience with them as well. However, I quickly discovered that handling user input complicated it greatly and required many hours to figure out. I learned about regular expressions to correctly split up the user input into "command", "line number" (if applicable), and "text to insert". I also learned about vectors of string to contain user input, and now I understand the benefits of vectors in C++. So I suppose I should be thankful for learning new types of coding techniques that I'm sure will be very useful in the future. My actual implementation of the linked list involved a simple Node struct coupled with a more complicated Linked List class that used vectors, both of which were functional and served their purpose. At first, I also had an Editor class that called upon the Linked List class, trying to achieve good encapsulation, but in the end I realized the Editor class was redundant and got rid of it. In hindsight, using a Template might have been more efficient or at least more applicable for future use because it is more generic and probably would have been less complicated. At first, I didn't understand why I needed to implement an entire Linked List class since a simple Node struct was sufficient for accomplishing most of what I needed, so this brought up the discussion of private variables versus public access modifiers. I considered adding more data to the Node structs by giving them a dataSize attribute of type int besides the next pointer and value attribute; however, once again I realized this was redundant. Another noteworthy comment is that the instructions involved outputting no error messages when the user enters invalid input for a command or text, and I found this troublesome for a variety of reasons. First and foremost, it made debugging more difficult, so I actually incorporated my own error messages and exceptions when I was first writing my code. In reality, not having any error messages is not user-friendly and would generate some serious complaints. If I could do things differently, I definitely wouldn't have tried to over-complicate things with various classes and vectors and regular expressions, because it was so complicated I couldn't even figure out how to fix the line numbers when I deleted the node. Overall, I am very fed up with this project and don't want to think about it for a very long time, but at least now I have a good understanding of Linked lists.