Commentary

This line editor implemented through a linked list was a fairly complex assignment in that it made me think of the data structure and user input in a completely different way than I perceived it before. Analyzing each method for it's time complexity, I came to the conclusion that insert was O(n) or O(1), insert end was O(n) or O(1), deletion was O(n) or O(1), edit was O(n) or O(1) and search was O(n) or O(1). The reason that I came to the conclusion that each time complexity was O(n) was that (assuming you aren't using an iterator), you must loop through to the deserved node each time you wanted to do something. The advantages for using a linked list as a line editor is that each sentence flows into another so it would be logical to access these values in order. The disadvantage is that if you wanted to change or edit any single value that isn't the first line, you have to iterate to the line that you want sacrificing time, where as if you used an array, it would only take O(1) time. I learned how linked lists worked on a deeper scale through this project as well as how time complexity comes into play. If I were to do this assignment again, I would work better with the way I keep track of line numbers because the way I implemented that was awfully, inefficient and could have been done more clearly.