Programming Assignment 1 Commentary

In my implementation of a linked list, I have six distinct methods: insertEnd(string), insert(string, int), find(string, int), find(string), deleteEntry(int), and print(). All of these methods have a computational complexity of O(n), since they need to traverse, in the worst case, the entire linked list.

The advantages of using a linked list as a line editor are that the linked list can be dynamically resized by adding references to successive nodes, that is, the length of a linked list does not have to be determined before the start of program execution. Also, when insertion and deletion are performed, a linked list does not have to move elements left or right like in a static array; only the references to two nodes in two node of the linked list have to be changed. Linked lists also have a low computational complexity for their methods, O(n). However, disadvantages of using a linked list include no indexing of elements. Since linked list elements are not indexed, methods that rely on an index, like editing, require traversing the linked list to find the node whose text needs to be edited. If an array were used to implement the line editor, editing would have O(1) complexity instead of O(n) complexity, since editing takes in a line number which refers to a specific index. Overall, I think a linked list is a good implementation of a line editor because, despite the increased computational complexity of editing, insertion and deletion are much easier in a linked list since only references to the next node for two nodes need to be changed, not the placement of successive elements as would be the case in an array.

From this assignment, I learned how to properly implement a linked list and how to parse command line input that does not have the same number of words or numbers every time. If I had to start over, I would have made my loops more consistent, that is, not using some while loops and some for loops for traversing the linked list, just a consistent loop structure for each operation. With this consistency in traversing operations, I would optimize my linked list code to reuse the same traverse operation in each method, where I would have the traversing method return the specific node whose references or attributes would be changed depending on the command specified.