Ashley Rocha

COP3530: Data Structures & Algorithms

Assignment 1: Line Editor – Commentary

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

> The computational complexity for method in which a new node is inserting at the end is O (1) because in my implementation of the linked list, I used a deque, which holds the nodes. Adding a value to the end of a deque has constant computational complexity. The other methods, such as inserting at a particular index, searching, printing, editing, and deleting have a computational complexity of O (n) because in order to accomplish these tasks, it is necessary to traverse the list n times.

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

> Using a linked list was an interesting way to create a line editor. Some advantages I found were when printing, it was very easy to implement the line numbers and consecutively print the list. Also, deleting lines was easy to accomplish by updating the neighboring nodes. Disadvantages of using a linked list for implementing a line editor, include the time it takes delete data. Using the deque's erase modifier, if not at the beginning or end, this process has a complexity of O(n). Also, with our implementation it would be difficult to delete just one word in a line, a user would have to edit the entire line with the updated changes. For a document with shorter lines this is not as troublesome, but as the lines grow in length, this can take a lot more time and effort.

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

> I learned how to implement linked lists and an application of them that I had not thought of. If I had to start over I would try to better plan out my program better before starting, rather than jumping right into coding. With better planning in the future, I anticipate that this may help decrease the amount of time spent debugging. Also, I would try to think of additional unique test cases to make sure my program is able to handle a variety of input, good or bad. In the sample cases, and cases I tried myself, I did not have any issues, however in Stepik for the last test case (#13) my program had a runtime error and I could not find what was causing this to happen. This also ties into planning beforehand, if I were to start over I would try to pay better attention to different cases my methods would have to deal with and account for each to prevent problems such as the runtime error that was occurring.