Alejandro Pineda

09/21/18

Programming Assignment #1 Commentary

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

Due to the fact that that in each method we have to transverse through the linked list there is a computational complexity of O(N) even though accessing a node is usually O(1).

insertEnd – O(N)

insert – O(N)

delete – O(N)

edit – O(N)

print – O(N)

search – O(N)

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

I thought that using a linked list for implementing a line editor had both its advantages and disadvantages. Some of the advantages of using a linked list is that each node had its data value, the string, and that we can transverse through the nodes and have access to this string. Unlike stacks and queues that can only access the head, linked lists can transverse and access the whole list. Another advantage is that a linked list does not have a set size, this was useful since we could always insert at the end or somewhere in the middle. Furthermore, another advantage of using a linked list you can completely delete nodes. In arrays, we can only delete what is inside a specific index but the size does not change. On the other hand, when we use linked list we can completely get rid of a node and reduce the size of the linked list. Last but not least, I think that having to create our own linked list class was advantageous because we could design it to work for our methods that we had in mind as well as built in the methods around out linked list implementation.

I believe that there is only one clear disadvantage when using linked lists which is that there is no built-in indexing. When using linked lists, we must transverse the whole list

to find a node, this leads to a greater computational complexity when searching for a node since you must transverse the whole linked list.

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

This assignment was very beneficial to me, I felt like I learned and practiced a lot of important data structures concepts. I practiced how to transverse through a linked list as well as how to modify it. Specifically, I practiced how to add a node at a specific spot as well as how to delete which we had already done on Stepik. Last but not least, I had more exposure to some C++ methods such as stoi and getLine(). If there was something that I would differently would be to start the project earlier and find a simpler method for separating the command, text and index, I used substrings and find() and that was often confusing and hard to keep track of.