Abby Pham

Programming Assignment #1 Commentary

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

The methods in this implementation and their complexities are:

- For insertEnd, the computational complexity of my method is O(1) constant time. This is because my “tail” node at the end of my linked list makes it very easy to simply link the new node to the end of the linked list.
- For insert, the complexity may depend on the desired index of the new node. If the node is to be added at the first index, the new node can replace the head of the list quite easily and be O(1). However, for adding to the middle of the list and at the end of the list my method has to traverse through some of all of the list and thus makes the complexity O(n).
- For delete, the complexity also may depend on the desired index of the node that is being removed. If the first line is being deleted, the complexity is O(1) because the head is moved up to the next node without needing any traversal. If the line is in the middle, the complexity is O(n) because the method needs to go through the list and iterate to the proper node. Deleting the node at the end also requires O(n) time because the “prev” node must be traversed to, as the second to last node.
- For edit the complexity is O(n) due to it having to iterate to the desired index at anywhere in the list to change the text.
- For print, the complexity is also O(n) because every node in the linked list must be printed.
- For search the complexity is O(n) linear time because every line in the list must be searched and printed if the given text is found.
- My helper method findLength() is also O(n) because it counts the number of nodes in the list and thus must iterate n times.

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

When using a linked list for this line editor, the disadvantages were associated with accessing different nodes and searching. Every time a specific node was to be manipulated or found, a new pointer would have to be instantiated and iterated throughout the whole list. In terms of advantages, it was relatively easy to add or delete lines from the top or bottom of the list.

Abby Pham

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

If I had to start over, I would implement a doubly linked list instead of a singly linked list. I think my program could have benefitted from having a “previous” node pointer along with the “next” pointer it already has. The “tail” node I had along with my “head” node for my list helped tremendously, as it seems the more pointers and references you could have, the easier the implementations were and the less complex.