David Espantoso
4481-5925

COMMENTARY: PROGRAMMING ASSIGNMENT #1

Programming Assignment #1 prompted students to implement a line editor using the linked list data structure. In my program, I coded methods to insert at the end, insert at a specific index, delete at a specific index, edit at a specific index, print, search for a specific text, and quit the program. The way my line editor work was by using a Singly Linked List and each node of the linked list had a value of data type string, which was the text that the user would input. The computational complexity of my insertEnd, insertAtIndex, deleteAtIndex, searchText, and printList methods are all constant time or O(1), at best case. insertEnd will always be O(1) as we have a tail pointer that allows constant insertion. However, the worst case for insertAtIndex, deleteAtIndex, and searchText is linear time or O(n), which means that the operation traverses through the whole list. These computational complexities is what makes using a linked list for a line editor an ideal data structure. Using a linked list for implementing a line editor is very beneficial, because mostly every operation ran at constant time. Using a linked list allows the programmer to track and consider all lines and their respective values. By making each line a node in the linked list, it makes doing operations on the line editor easily managed. Using a linked list did not provide any disadvantages while implementing the line editor, as everything was organized and useful. From this assignment, I gained a better understanding of how linked list work. I started off using a Doubly Linked List, in order to be able to track my next node and previous node, however, I learned with time that I was complicating myself, as working with a Singly Linked List would be more efficient. Also, I began doing this assignment early, in contrast to previous semester, where I would procrastinate on programming assignments. By starting my assignment early, I was able to code with ease and solve problems carefully. I was not speeding through my code and doing this helped foster a better understanding of using linked lists.