

Paul Calzada

9-21-18

Programming Assignment 1

Data Structures and Algorithms

Professor Cheryl Resch

Commentary

1. The computational complexity of the methods in the implementation are described in the table below:

Function	Computational Complexity
InsertEnd	$O(1)$ (using tail variable)
Insert	$O(n)$
Delete	$O(n)$
Edit	$O(n)$
Print	$O(n)$
Search	$O(n)$

2. I am interested in how we used a linked list for a line editor. The largest advantage here is the linear computational complexity for inserting and deleting. The largest disadvantage I can see is accessing the data, which is also linear. Arrays can access data at constant complexity. Because the line editor frequently needs to access data at each line to either edit or search or print, arrays might be useful here. However, there are drawbacks, as their length is fixed. Even vectors are slow because as they grow, they get fragmented in memory and become cumbersome to manage. The linked list seems like the better option in the end because of its flexible length, its linear insertion and deletion time, its constant insertion time at the end of the list, and its ability to use different data types. For a line editor, a linked list is a great choice.
3. I learned the true application of data structures in a tool that I personally use often. I also rarely use linked lists in personal projects or school projects but am more inclined to do so now. I would do many things differently in the future. I would probably create an `isEmpty()` function and other helper functions to ease much of the tedious tasks when accessing elements from the linked list. I also might try using an iterator next time which might ease many of the processes. I also intend to comment as I code rather than comment after I code. These are only a few that I had in mind to develop myself and my coding skills in the future.