Paul Kenaga

Professor Ostrowski

CS 300

5 Nov. 2021

Linked List Code Reflection and Flowchart

The code for this assignment demonstrates append, delete, prepend, and search operations executed on a linked list that was created from a CSV file of government bids. The time complexity for deleting and searching for an element is dependent on the size of the list and is reflected as O(n) in big O notation because the list must be traversed until an element with a matching bid ID is found. Append and prepend have a constant big O notation, O(n), because we can easily reference the head and tail nodes to change them and traversing a list is not necessary. This code also traverses the list to execute a print method that displays all of the bids. Loops used in this code begin at the head node and execute until “nullptr” is reached, either as the current node or the next node depending on the operation.

I enjoyed working on this assignment because it gave me a better understanding of how to use a linked list and what sets it apart from other data structures with regards to runtime complexity and operation implementation. I ran in to an issue during the initial development of my code with the Remove method. After removing an element, selecting either the display all bids or the find bid option would cause the program to end. I found that changing the loops I was using to compare bid IDs for the remove and search methods resolved this issue.

