Jarrod Helmers

CS-300

Dr. Webb

23 March, 2024

Module Three Assignment: Linked List

I coded a linked list data structure in C++ to manage bids. The code performs operations like adding, removing, searching, and displaying bids. Developing the code was challenging, especially dynamic memory allocation and handling pointer manipulation. Memory deallocation during bid removal was a primary issue, which required special attention to updating pointers and freeing memory. Implementing search functionality required careful traversal of the list. This experience provided valuable insights into data structure implementation and memory management in C++.

Pseudocode:

Start:

Create LinkedList class with head, tail, and size variables

Create Node struct with bid and next variables

LinkedList():

Initialize head, tail, and size variables

~LinkedList():

Delete all nodes in the linked list

Append(bid):

Create a new node with the bid

If list is empty, set head and tail to the new node

Else, make the current tail node point to the new node and update the tail

Prepend(bid):

Create a new node with the bid

If list is empty, set head and tail to the new node

Else, make the new node point to the current head and update the head

PrintList():

Start at the head

While current node is not null, print bid information and move to the next node

Remove(bidId):

If head node matches bidId, remove head node

Else, start at the head and loop over each node

If next node matches bidId, remove next node

Else, move to the next node

Search(bidId):

Start at the head

While current node is not null, check if bidId matches current node's bidId

If match found, return the bid

Else, move to the next node

If no match found, return empty bid

Size():

Return the size of the linked list