Benjamin Leanna

CS300 DSA – Week 4 Hash Table pseudocode and reflection

01/27/2023

This week we were to make Hash Table functions work in our program to be able to do 4 things: load bids, display all bid, find bids, and to remove bids from a CSV file. Issues that arrived came in the form of complicating where pointers were going to and from in the remove process. The video released sort of explained how that worked. What I mean is when removing a bid, we needed to change the previous pointer to the current one and then delete the current one and make the current now point to the next. This was a tricky thing but after watching the lecture video, made it work.

HashTable class’s HashTable function

Initalize node structure by resizing table size

HashTable class’s HashTable function’s size // used to improve efficiency of hashing algorithm

Invoke local table size to size with ->

Resize nodes size

HashTable class’s HashTable destructor

Erase nodes at the beginning

HashTable class’s hash key function

Return key table size

HashTable class’s Insert function

Create the key for the given bid

Retrieve node using key

If no entry found for the key

Assign this node to the key position

Else

if node is not used

Assign old node key to UINT\_MAX

Set to key

Set node to bid

Set next node to nullptr

Else find the next open node

While node next is not equal to nullptr

Node equals the next node

Next node equals a new node from bid and key

HastTable class’s PrintAll function

For node begin to end iterate

Output key, bidId, title, amount and fund

Node is equal to next iteration

While node not equal to nullptr

Output key, bidId, title, amount and fund

Node is equal to next iteration

HashTable class’s Remove function

Set key equal to hash atoi bidId c\_string

If the node key is not equal to UINT\_MAX

Remove the node

If the next node is equal to nullptr

The key node will equal UINT\_MAX

Else

The node at the key will equal to the next pointer

Else

Make current pointer to next

Make previous pointer to the node

While current pointer doesn’t equal nullptr

If the current bid’s ID is true

Previous node equals to current

Delete the current

Current is now nullptr

Return

Assign the pre pointer to the current

Delete the current pointer

Current pointer is now nullptr

Return

Assign the previous pointer to the current pointer

Current pointer now points to the next pointer

HashTable class’s Search function

Create new bid

Create the key for the given bid

If entry found for the key

Return the node bid

If no entry found for the key

Return the node bid

While node is not equal to the nullptr

If the current node matches

Return the bid

Node is equal to next node now

Return the bid