Jarrod Helmers

CS-300

Dr. Webb

30 March, 2024

Module Four Assignment

The following C++ code implements a hash table that stores bid information using chaining. The HashTable class includes methods for inserting, printing, removing, and searching bids based on their unique identifiers. Each bid is represented by a struct that includes details such as bidId, title, fund, and amount. The hash table employs a vector of linked lists to handle collisions, ensuring efficient storage and retrieval of bid information.

I faced a challenge with deallocating memory in my project's destructor. I revised the destructor to delete only dynamically allocated nodes to ensure memory safety. The hash table with chaining implementation for storing bid information is modular, efficient, reusable, and meets all the specified requirements.

Class HashTable:

Data Members:

- nodes: vector of Node structs

- tableSize: unsigned int

Methods:

- Constructor (HashTable()): Initialize nodes vector with default size and tableSize to DEFAULT\_SIZE

- Constructor (HashTable(size)): Initialize nodes vector with specified size and set tableSize to size

- Destructor (~HashTable()): Clean up dynamically allocated memory by deleting nodes

- hash(int key): Calculate the hash value of a given key using modulo operator

- Insert(Bid bid): Insert a bid into the hash table

Class Node:

Data Members:

- bid: Bid struct

- key: unsigned int

- next: pointer to next Node

Methods:

- Constructor (Node()): Initialize key to UINT\_MAX and next to nullptr

- Constructor (Node(bid)): Initialize bid, key to UINT\_MAX, and next to nullptr

- Constructor (Node(bid, key)): Initialize bid, key, and next to nullptr

Pseudocode for Insert(Bid bid) Method:

Method Insert(Bid bid):

1. Calculate the hash key for the bid using the hash function.

2. Retrieve the node using the key from the nodes vector.

3. If no entry is found for the key:

a. Assign the bid to the node and set the key to the calculated key.

4. Else:

a. Traverse the linked list starting from the retrieved node until reaching the end.

b. Create a new Node with the bid and key and set it as the next node.