Prathyoosha Chaya

CS 163 DATA STRUCTURES

Prof. Karla Fant

**DESIGN WRITE-UP 4:**

**HEADER FILES:**

The header file will include the table class, and node struct definition, with the function prototypes of the following public/private member functions.

TABLE HEADER FILE: This header file includes the BST class along with its table ADT operations (and their wrapper functions), the node structure (left, right, and data pointers), CS term data structure (name of term and definition). Its private members include a pointer to the root of the tree, file IO variables, and the private utility functions to convert the search term to be used in retrieval and removal.

Constructor: The purpose of the constructor function is to initialize the data members. Here, the pointer to the root of the tree will be set to null. In the constructor for the term data structure, the pointers to the term name and definition dynamic arrays will be set to null.

Destructor: The destructor will deallocate all previously allocated dynamic memory - the pointer to the root of the tree will be deleted, in the case of the BST class. In the node structure destructor, the left and right pointers and the pointer to data will be deleted (with array subscript for data pointer).

Insert term function: The insert term function is public and takes the data of the entry to be inserted (provided by client) as a parameter, along with a pointer to a node. It returns an integer value depending on the case of the insertion success/failure by returning the count of nodes traversed until insertion. It will first compare the term name of the user’s data to the root (or sub-root, depending on the recursion state), then go left or right of the term is less than root, or greater than or equal to root. It will call the recursively call the insert function with the left and right pointers.

Remove term function: The remove term function is public and takes a pointer to a node as a parameter. It first displays the all of the CS terms. The user enters the term that they would like to remove (which runs through the convert to uppercase function), which is then used as a comparison term when traversing the tree. If the item is found at a leaf, then it is deleted and set to null. If it is an internal node with 1 left or right child, then a temp pointer is created to point to its child and the root is deleted and set to the temp pointer. If it is an internal node with 2 children where the right node has no left children, then the right child’s data is copied to the root’s data. If the right node has a left child, then the tree is traversed until the left pointer is null, dragging a prev. pointer, then the parent adopts the temp’s right child and the in-order successor’s data becomes the root’s data. The in-order successor is then deleted. If a match to the search term is not found or the tree is empty, nothing is removed. It returns a success or failure flag of if the item was removed/not found/not removed.

Retrieve function: The purpose of the public retrieve function is to take the search term to be found from the client program and return the data of that term (if it is found) as a parameter. It uses the same algorithm used in the insert function, and returns the found item’s data through the function parameters. If a match to the search term is not found or the tree is empty, no data is returned as a parameter. It returns a success or failure flag of if the item was removed/not found/not removed.

Display all function: This public function will display all of the information in the archive (table) by going to the root node in the tree and displaying, then returning the function calls for the left and right subtrees. It returns the success/failure case (if the table is empty) by Boolean value, and it takes a pointer to a node as a parameter.

Display range function: This public function will display a range of information in the archive (table) by using the same traversal algorithm as the insert function to find the lower limit of the display. It then proceeds to recursively call the display function for all rightward subtrees upon the return call. It returns the success/failure case (if the table is empty) by Boolean value, and it takes a pointer to a node as a parameter.

Get height function: The public get height function takes a pointer to a node in the tree as a parameter, and returns the height of the tree. It returns and integer value of the max value of the height function called with the node left of root + the height function called with the node right of root +1.

Is efficient function: The public “is efficient” function determines the efficiency of the tree by calculating the balance factor of the left and right sides of the tree. The balance factor is calculated by subtracting the right side of the tree’s height from the left side’s height, with a perfectly balanced tree having a balance factor of 0. This is done by calling the height for the root’s left and for root’s right, then subtracting them. The value returned is a Boolean value of if the height is 0 or not.

Write function (to accompany load function): The purpose of this function is to write the contents of the tree back into the file. It has no parameters. The entry data will be outputted to the file (no append), separated with “:” delimiters. At the end of the CS term, a “\n” delimiter is placed and the index is incremented. This is recursively called until all tree members are written. It returns a failure or success case.

Load function: The purpose of the load function is to take the CS term information in the text file and store it into the tree. It first creates temporary arrays to store the term name and opens the text file to attempt to write the term to set EOF. While the file has not reached the end, it compares each value to the current subtree’s root (done recursively) – if it is greater than or equal to the root, it enters the right, and if less than root, it enters left. It uses the existing delimiters mentioned in the “write” function to read into each entry’s fields.

Convert to uppercase function: The purpose of this private function is to convert the inputted array of characters to uppercase, using the toupper functionality in the cctype library. It takes a source and destination array of characters as input parameters, and changes the contents of destination to be the uppercase version of the source array of characters, going character by character. This will be called in the retrieve and remove function.

IMPLEMENTATION FILE:

The first implementation file will contain the function definitions below.

In the main function, the menu interface will be implemented for the user to interact with the archive of CS term data. An object of type table “dictionary”, and an object of type cs term “user\_term” is created for the purpose of calling table class functions and passing the user’s information to be added into these functions. Then, the input file and output file variables (ifstream/ofstream type) are created, and a “menu option” variable (type int, to keep track of the menu item entered). The text file is opened, and the load function is called.

Then, the menu is outputted by calling the menu function. The user can choose which option they would like to execute from the following menu format:

1: Display all information.

2: Display all information in a range.

3: Add dictionary item.

4: Remove dictionary item.

5: Retrieve information.

6: Get height of dictionary.

7: Check efficiency of dictionary.

8: Save and quit CS dictionary.

9: Quit CS dictionary.

The user will be prompted to enter a number, which is stored into the “menu option” variable once the function exits. If a 1 is entered, then the “display” function is entered. At this point, all data from the text file will have been read into the tree based on its alphabetical order by term name. Once the function is exited, the menu function is entered again. The functions continue to be called until the user decides to quit (or save and quit). The program can be quit without writing to the file, or quit after writing the contents of the table to the file.

Menu function: This function returns an int value. It outputs the menu for the user to select, and returns the user’s selection. The menu displayed is shown in the main function.