Assignment 4

Trees and AVL Trees

Data Structures (Section A & B), Fall 2018

The Assignment 4 consist of two parts and each part have different deadline.

Late submissions will not be accepted.

Due Date Assignment 4 PART 2**: 12:00 pm, 6th November 2018**

Submission Location: Upload the zip file containing your solution on the Google classroom. The name of the ZIP file should be your Roll Number.

ASSOCIATIVE MAPS: Associative Maps or Tree Maps are an important data structure that have many uses in real-world. Conceptually, Map is a data structure that consists of are collection of **key and value pairs.** In a map, the *key values* are generally used to sort and uniquely identify the elements, while the *mapped values* store the content associated to this *key.*

Problem Statement: Consider an online shopping mart. The online shopping marts advertise and display million of the products on their website. However, they do not have all the product in stock. The products that are most frequently purchased by customers are kept in warehouse while the rest of the products are bought when a demand is placed by a customer.

We wish to keep track of the most frequently bought products along with their name and quantity on hand (quantity available in ours stock). So, whenever a customer placed an order we can quickly check if the product is available in our stock or not.

Conceptually, we want to have an associative map of products:

Key (product id)	**Value (product name, quantity available in stock)**
123	**Name**: AMD GPU **Quantity:**5
2098	**Name:** Intel Corei7 CPU **Quantity:**10
567	**Name:** Kingston RAM **Quantity:**350
39911	**Name:** DDR3 RAM **Quantity:**50

PART 1) Implement **Associative MAPS** using **AVL TREES:**

1. **Provide following operations in your BST class**
   - **a) Insert:** Provide **Recursive** Insert function. The Insert function will insert a given value in AVL MAP. Incase value already exist then it should return false and do not insert.
   - **b) Delete:** Provide **Recursive** Delete function. The delete function will delete the node with given key. Incase key does not exist then it should return false. Your code will perform deletion by copy.
   - c) **LookUp:** Provide Recursive function to find if the record with the given Key exists.
   - **d) PrintINORDER: Recursively** Print the records sorted by the key.
   - **e) Destructor**

f) PRINT the AVL Tree on Screen in the following manner

```
          50
         /  \
       20    70
      / \      \
    10   30     85
```

Only print keys (do not show values)

You can use the following function to move cursor to (x,y) coordinate on the screen

```
#include <stdio.h>
void PlaceCursor(int x, int y){
        COORD c;
        c.X = x;c.Y = y;
        HANDLE h = GetStdHandle(STD_OUTPUT_HANDLE);
        SetConsoleCursorPosition(h, c);
}
```

If you can find the height of the tree and the number of leaf nodes, then you can easily print the tree.

2. **Load: create an AVL** from the file data. The file will consist of ID, Name and Quantity. File format will be as follows (the fields are separated by tab) (not in BST class)

ID	Name	Quantity
123	AMD RAM	75
245	K RAM	47

AVLMAP

```
template<class K, class T>
class AVLNode {
public:
            K Key;
            T * data;
            int height;
             AVLNode<T> *left, *right;
};
```

```
template<class K, class T>
class AVLMAP {
/* Note we are making AVLMAP not node*/
  /*Add all the functions here */
private:
            AVLNode< K, T >* root;
}
```

```
class ProductData {
public:
/* Provide all required operations here*/
/* Overload all the required operators needed for the proper functioning*/
private:
             string name; //Value
             int quantity; //Value
};
```

```
int main() {
    cout << "Associative Map using AVLMAP";
    AVLMAP<int, ProductData> ProductMap;
    return 0;
}
```

NOTE: Do not create an object of PRODUCTDATA in any function of AVLMAP class.

AVLMAP should not know about the PRODUCTDATA class.
AVLMAP will compare data of different nodes using the key store in it.
Overload all the operators that you need in PRODUCTDATA class.

Which implementation of Associative Map is better? Maps using AVLMAP or BST. Why?

CODE DESIGN GUIDLEINES

- Do template-based programming
- Code should be properly intendent and commented (2 marks for this)
- Make sure there are no memory leaks or dangling pointers
- Don’t cheat or take too much unnecessary help from your friends