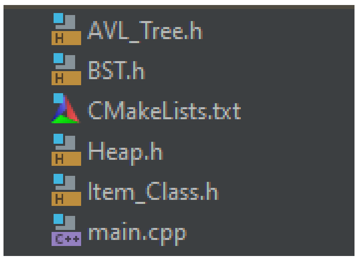
Data Structure - Assignment #2

Team Members

- **Youssef Mohamed Aboelfotoh** (20221202)
- **Mazen AbdelFattah Omar** (20230607)
- **Mariam Amro Ahmed** (20221217)
- **Basel Osama** (20221254)
- **Abdallah Gamal Mostafa** (20221238)

Overview

This document provides a brief overview of the code implementation for our data structure assignment. The project includes four header files for the AVL tree, BST (Binary Search Tree), HEAP, and item class.


AVL_Tree.h
BST.h
CMakeLists.txt
Heap.h
Item_Class.h
main.cpp

Item Class and Comparators

In the item class, we have implemented several comparators to be used by the trees. These comparators ensure that the item class fits appropriately into the data structures.

```
struct ComparatorbyPrice {
    bool operator()(const item &a, const item &b) const {
        return a.getPrice() < b.getPrice();
    }
};

struct ComparatorbyName {
    bool operator()(const item &a, const item &b) const {
        return a.getItemName() < b.getItemName();
    }
};

struct ComparatorbyCategory {
    bool operator()(const item &a, const item &b) const {
        return a.getCategory() < b.getCategory();
    }
};
```

Main.cpp and Menu Options

In `main.cpp`, we have created a mini-menu for interacting with each tree. Ensure that all headers are included appropriately.

Run the main program and use the menu options as follows:

1. **Add Item**
2. **Add Items from a File**
3. **Remove Item**
4. **Display All Items**
5. **Display All Items Sorted by Name Ascending**
6. **Display All Items Sorted by Name Descending**
7. **Display All Items Sorted by Prices Ascending**
8. **Display All Items Sorted by Prices Descending**
9. **Output All Results to output.txt**
10. **Return to Main Menu**

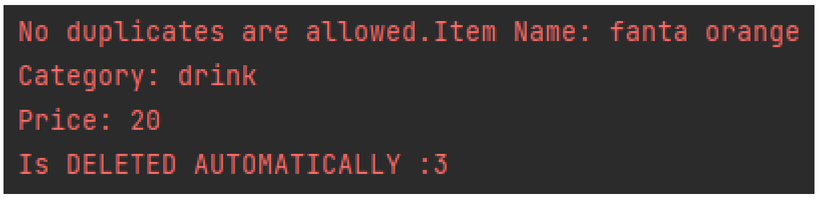
Note: The "Add Items from a File" option takes input from a file instead of the console.

Options 4 to 8 display the results in the console. Outputting all items to `output.txt` provides a comprehensive result for options 4 to 8, allowing you to see all sorted data at once.

Important Note on Duplicates

In AVL and BST trees, duplicates are not allowed. For example, in a tree sorted by price, if both Pepsi and Fanta have the same price, the program handles this by refusing to insert Fanta. Neither an AVL tree nor a BST can have duplicates.

However, in a tree built according to name (not price), duplicates are allowed, meaning both items can coexist in the tree.



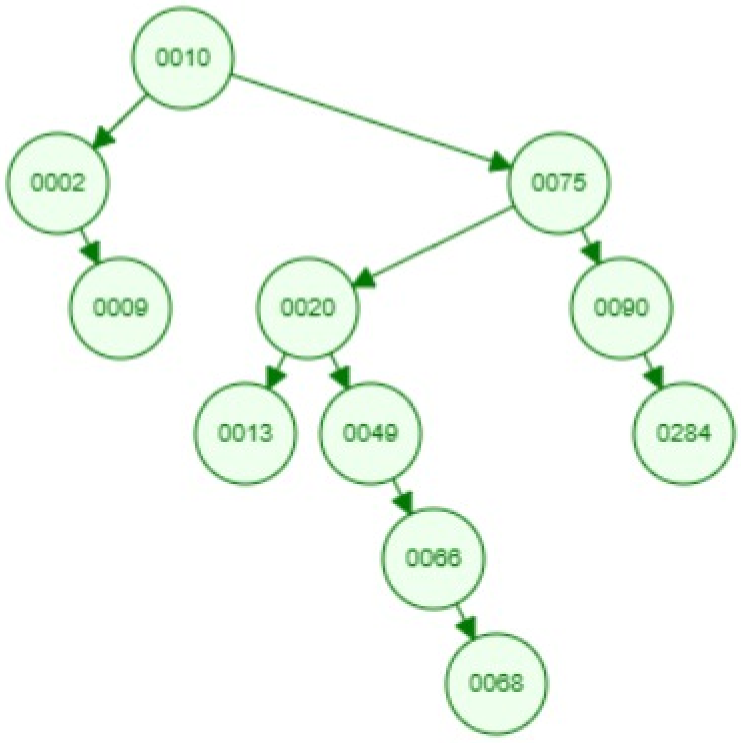
```
No duplicates are allowed.Item Name: fanta orange
Category: drink
Price: 20
Is DELETED AUTOMATICALLY :3
```

All of our Trees were tested on websites provided by doctor and we have the same results

https://www.cs.usfca.edu/~galles/visualization/AVLtree.html

https://www.cs.usfca.edu/~galles/visualization/BST.html

https://www.cs.usfca.edu/~galles/visualization/Heap.html


0010
0002
0075
0009
0020
0090
0013
0049
0284
0066
0068