---------- *Wheaton College, MA - Spring 2024* ----------

COMP 318 Algorithms - Project 2

Description - *AVL Trees*

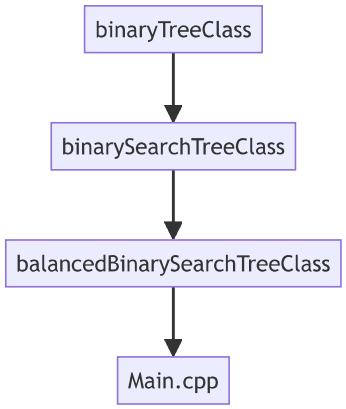
This project introduces an implementation of a Balanced Binary Search Tree (BST), specifically an AVL tree, which self-balances to maintain optimal search, insert, and delete operations times.The program offers multiple options to manage a balanced BST effectively, leveraging various algorithms and data structures, including tree nodes, rotation mechanisms, and balanceOurTree factor calculations.

Highlights of the project:

- **Data Insertion:** Safely inserts data into the tree, guaranteeing the maintenance of BST properties.
- **Tree Traversals:** Implements and demonstrates in-order, pre-order, post-order, and level-order traversals of the tree.
- **Tree Visualization:** Displays up to four levels of the tree, providing a clear visual representation of its structure.
- **Balance Factor Calculation:** Calculates and displays the balanceOurTree factors, crucial for understanding the AVL tree's balanceOurTree.
- **Single Rotations:** Executes single right and left rotations to maintain AVL tree balanceOurTree after insertions and deletions.
- **Double Rotations:** Performs left-right and right-left rotations to balanceOurTree more complex imbalances in the tree.

Developed by Sam Hammami '25

Inheritance diagram:



This project contains multiple files that divide the workload.

- main.cpp:
  This file is used to test the AVL Trees implementation. It creates a tree, inserts nodes, displays the tree, and deletes nodes. It also displays the tree after deletion.
- AVLtrees.cpp:
  This file is used to create the AVL Trees Object Data Structure which holds the nodes of the tree and the methods to manipulate the tree.
- AVLtrees.h:
  This is the header file for the AVLtrees.cpp file. It contains the three classes hierarchy (binaryTreeClass, binarySearchTreeClass, balancedBinarySearchTreeClass) and the methods' prototypes.

Getting Started

This program uses the following libraries:

- #include <iostream>
- #include <queue>
- #include <iomanip>
- #include <cmath>

CMake Minimum version

- cmake_minimum_required(VERSION 3.27)

Compiler version

- Clang C++ compiler

Installing /compiling

On CLion

After downloading the code, please create a new project on CLion and select the folder containing the program as the project folder. It will ask you to either use the files existing inside the folder or create a blank design, please select to use the files inside the folder. The CMakeList.txt file and cmake-build-debug folder should include all the necessary files needed to run this program on CLion.

Executing program

On CLion

Click the run button - or - Shift + F10

Run the code - It should be like:

```
.                       ***     Welcome to The AVL Trees World     ***                    .


-----------------------------------------------------------------------------------------------

Inserting: a b c d e f g h i j k l m n o p q r s t u v w x y z

Loading the tree...

                    z
                  /
                y
              /
            x
              \
                    w
                  /
                v
                  \
                    u
          /
        t
          \
                s
              /
            r
              \
                q
 /
p
 \
                    o
                  /
                n
                  \
                    m
              /
            l
              \
                    k
                  /
                j
                  \
                    i
          /
        h
          \
                    g
                  /
                f
                  \
                    e
              /
            d
              \
                    c
                  /
                b
                  \
                    a

Number of nodes: 26
Height of the tree: 4
Balance Factors: p:0 h:0 d:0 b:0 a:0 c:0 f:0 e:0 g:0 l:0 j:0 i:0 k:0 n:0 m:0 o:0 t:-1 r:0 q:0 s:0 x:0 v:0 u:0 w:0 y:-1 z
:0
```

```
-----------------------------------------------------------------------------------------------

Traversals:

Pre-order: p h d b a c f e g l j i k n m o t r q s x v u w y z

In-order: a b c d e f g h i j k l m n o p q r s t u v w x y z

Post-order: a c b e g f d i k j m o n l h q s r u w v z y x t p

Level-order: p h t d l r x b f j n q s v y a c e g i k m o u w z

Level-order by Sam:
p
h t
d l r x
b f j n q s v y
a c e g i k m o u w z



Searching for node 'c': Found
Searching for node 'm': Found


-----------------------------------------------------------------------------------------------

Deleting some nodes...

node 'h' - Deleted
node 'i' - Deleted
node 'm' - Deleted
node 'g' - Deleted
node 't' - Deleted
node 'a' - Deleted
node 'z' - Deleted
node 'x' - Deleted

Deleting node 'm' - Test when a node already do not exist


** after deleting - Display the new tree **

         y
           \
                  w
                /
               v
      /
    u
      \
               s
             /
         r
           \
              q
 /
p
 \
                   o
                 /
               n
             /
         l
```

```
          \
             k
      /
    j
      \
             f
               \
                  e
          /
        d
          \
                  c
               /
             b

Number of nodes: 18
Height of the tree: 4

Searching for the deleted node 'm': Not Found
Searching for the deleted node 'g': Not Found
```

Authors

List of authors/contributors' names and contact info:

- Sam Hammami '25 - hammami_houssem@wheatoncollege.edu

Version History

- Starter Code Uploaded - April 04, 2024
- CLion Set up - April 05, 2024
- Adjusted AVLTrees.h - April 05, 2024
- Completed 1st Class (BT) - April 06, 2024
- Some work on Main.cpp - April 06, 2024
- Planning Binary Search Tree Class - April 06, 2024
- Developed 2nd Class (BST) - April 07, 2024
- Started Implementing 3rd Class (B_BST) - April 07, 2024
- Checking Rotations (Single/Double) - April 09, 2024
- Discussing the project with Professor Tony - April 09, 2024
- Solved nodeType Typo issue - April 09, 2024
- Completed the 2nd Class (BST) - April 10, 2024
- Discussed my AVLtrees.cpp/.h with Peer Tutor.h - April 10, 2024
- Solved some issues & More work on Main.cpp - April 10, 2024
- Completed the 3rd Class (B_BST) - April 11, 2024

- Testing Main.cpp (Specific Cases, Display, Delete) - April 11, 2024
- Commenting the code (Sam style) - April 12, 2024
- Final Testing - April 12, 2024
- Checking-In with Professor **Tony** - April 15, 2024
- Uploaded_Final Version - April 15, 2024

License

Acknowledgments

- We acknowledge all the programming input from Professor **Tony Tong** put into this program.
- We also acknowledge **Sam Hammami** for his hard work and dedication to this project.