C++ Implementation of AVL Trees

Mingyu Guo

1 Task Description

You are asked to use C++ to implement

- Binary Tree Traversal

- AVL Tree Insertion and Deletion

2 Submission Guideline

You must follow this guideline! Your submission will be marked automatically. Failure to follow this guideline will result in 0.

Your submission should contain exactly 1 file: `main.cpp`.

You do not need to submit a design.

You should start your program by initializing an empty AVL tree. Your program takes one line as input. The input line contains n "modification moves" separated by spaces ($1 \le n \le 100$). The available modification moves are

- `Aint` (Character `A` followed by an int value between 1 and 100): `A3` means insert value 3 into the AVL tree. If 3 is already in the tree, do nothing.

- `Dint` (Character `D` followed by an int value between 1 and 100): `D3` means delete value 3 into the AVL tree. If 3 is not in the tree, do nothing.

Your input is then followed by exactly one finishing move (`PRE` or `POST` or `IN`): If the finishing move is `PRE`, then you should print out the tree (in its current situation) in pre-order. If the tree is empty, print out `EMPTY`. Otherwise, print out the values separated by spaces. `POST` and `IN` are handled similarly.

You don't need to worry about invalid inputs.

Sample input 1: `A1 A2 A3 IN`

Sample output 1: `1 2 3`

Sample input 2: `A1 A2 A3 PRE`

Sample output 2: `2 1 3`

Sample input 3: `A1 D1 POST`

Sample output 3: `EMPTY`

3 Websubmission

You are asked to submit via the web interface `https://cs.adelaide.edu.au/services/websubmission/`. The submission steps should be self-explanatory. Simply choose the correct semester, course, and assignment. The websubmission system will automatically fetch the latest version of your work from your SVN repository (you may also choose to submit older versions). Once your work is submitted, the system will launch a script checking the format of your submission. Click "View Feedback" to view the results. Your mark will

be calculated offline after the deadline. You are welcome to resubmit for as many times as you wish (before the deadline).

We will compile your code using `g++ -o main.out -std=c++11 -O2 -Wall main.cpp`. It is your responsibility to ensure that your code compiles **on the university system.**[1]

---

[1] g++ has too many versions, so being able to compile on your laptop does not guarantee that it compiles on the university system. You are encouraged to debug your code on a lab computer (or use SSH).