

Implementation of AVL Trees

1 Task Description

In this assignment, you have the option to choose C++, Python, or Java for your implementation. Your task is 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 one file, which should be named according to the programming language you choose: `main.cpp` for C++, `main.py` for Python, or `main.java` for Java.

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 ($1 \leq n \leq 100$). 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 from 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 do not 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 Marking

4 Submission Instructions

You are asked to submit via **Gradescope** <https://www.gradescope.com/> (by either direct upload or via a GitHub repository).

You are welcome to resubmit as many times as you wish before the deadline. We will compile and run your code using the specific commands for each language. The specific compilation commands used by the autograder are:

- For C++: `g++ -std=c++11 -o main.out -O2 -Wall main.cpp`
- For Python: `python3 main.py`
- For Java: `javac main.java` followed by `java -cp . main`

It is your responsibility to ensure that your code compiles and runs correctly on the Gradescope system, as compiler versions and environments may vary (e.g. g++ has too many versions, so being able to compile on your laptop does not guarantee that it compiles on the Gradescope system.)