POTD34.1. AVL Trees 3: Balancing an Unbalanced Node

Download and Extract

An initial setup of files is provided to you via a shell script: Download potd-q34

Using a terminal, extract the initial files by running the shell script you just downloaded (you will need to navigate to the directory where you saved the file):

```
sh potd-q34.sh
```

Your files for this problem will be in the `potd-q34` directory.

Balancing an AVL Tree

In a previous POTD, we were able to identify the deepest nodes that were unbalanced in an AVL tree. Now it is time for you to actually balance them!

Your task is to implement the following function

- `balanceTree(TreeNode*& subroot)` Returns which rotation type to use

A `RotationType`, which is defined in the `TreeNode.h` is simply an enumeration, which can take on the values:

- `right`
- `left`
- `rightLeft`
- `leftRight`

You are given the following helper functions

- `leftHeavy(const TreeNode* subroot)` Returns true iff the right subtree is higher
- `rightHeavy(const TreeNode* subroot)` Returns true iff the left subtree is higher
- `getHeight(const TreeNode* subroot)` Returns the height of the given subtree
- `printTreePreOrder(const TreeNode* subroot)` Prints the tree in pre order traversal

Given this unbalanced AVL tree:

```
      14
     / \
    11   22
   /  \
  7   13
 /
4
```

Calling `balanceTree` on node *14* should return the `right` rotation type because that is the correct rotation in order to balance the AVL tree, making it look like this.

```
      11
     / \
    7   14
   /   /  \
  4   13   22
```

Upload Solution

POTD 34

Total points:	0/1
Score:	0%

Question

Value:	1
History:	
Awarded points:	0/1

Report an error in this question

Previous question

Next question

Drop files here or click to upload.

Only the files listed below will be accepted—others will be ignored.

Files

TreeNode.cpp

not uploaded

Save & Grade | Save only