POTD32.1. AVL Trees: Finding Unbalanced Node

Download and Extract

An initial setup of files is provided to you via a shell script: Download potd-q32

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-q32.sh
```

Your files for this problem will be in the `potd-q32` directory.

The Problem

In order to balance a tree, we have to first find nodes about which rotations have to be made. In this question, complete the `findLastUnbalanced` function that accepts a `TreeNode *` root and finds the deepest node that is unbalanced. Remember that an unbalanced node has subtrees of heights differing by more than 1.

Note: If there are multiple unbalanced nodes, you have to return the one farthest from the root. If there are no unbalanaced nodes, return `NULL`.

Hint: You can use a helper function for calculating height.

Testing Your Code

In `main.cpp`, a simple test case has been created with the following binary search tree:

Example Input:

```
    8
   /  \
  5    13
 /\   /  \
4  7  10  14
      \
      11
       \
       12
```

Example Output: `TreeNode *` pointing to node *10*.

Upload Solution

Drop files here or click to upload.

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

Files

○ TreeNode.cpp

not uploaded

○ TreeNode.h

not uploaded

Save & Grade

Save only

POTD 32

Total points:	0/1
Score:	0%

Question

Value:	1
History:	
Awarded points:	0/1

Report an error in this question

Previous question

Next question