10520CS235100

Data Structures Homework #3

Due date: 2017/05/04 23:59

Upload .cpp and .h file to iLMS

Description

Please implement a tree (each node contains a 32bit signed integer) with six functions: “constructTree”, “deleteTree”, “treeHeight”, “treeWeight”, “leafNum” and “minPathWeight”.

1. **Node\* constructTree (Node \*root, std::string treeStr) const;**

A function can construct tree based on the **treeStr** which is an S-expression, and return a **Node\*** that points to the root of tree.

1. **Node\* deleteTree (Node \*root) const;**  
   Delete the tree and release the memory allocation of each node, then return a nullptr.
2. **int treeHeight (const Node \*root) const;**

Return the height of tree.

1. **int treeWeight (const Node \*root) const;**

Return the sum of node weights in a tree.

1. **int leafNum (const Node \*root) const;**

Return the leaf node number in a tree.

1. **int maxPathWeight (const Node \*root) const;**Return the min weight from root to leaf in a tree.

Implement these functions in **MyBinaryTreeOps.cpp** and **MyBinaryTreeOps.h**.

Don’t try to modify files in **readonly**, since we will replace them and use stricter testing.  
We provide a basic testing file 1.in in **readonly** and you can use it to evaluate correctness of your code.

Make sure your code can pass basic testing.

Note

Each of the 3 hidden test cases contains up to 10,000 trees and each tree may have 1,000 of nodes.

The time limit is “60” second.

Note: To get at least 60 points, you have to implement all the functionalities mentioned above. Always outputting the hard-coded results of the sample input without other implementations with get 0 points.

Input

The input is a S-Expression for a tree.  
E.g.:(99(5()())(35(-5()())()))

Output

If your code is correct, the output will be “[Accepted]”.

Pre-Correct 預批改

The deadline for pre-correct is 4/27 23:59.

4/27 23:59 未交的同學只是無法參予預批改而已，不會有任何影響。

想要被預批改的同學請務必要在4/27 23:59前繳交自己的code在ilms的hw3。

預批改的結果為「共對了?筆測資」以及錯誤型態，不會告知同學錯在甚麼地方。