HW3 Binary Tree

2016/11/15 10:00 a.m.

~

**2016/11/29 10:00 a.m.
(Hard Deadline)**

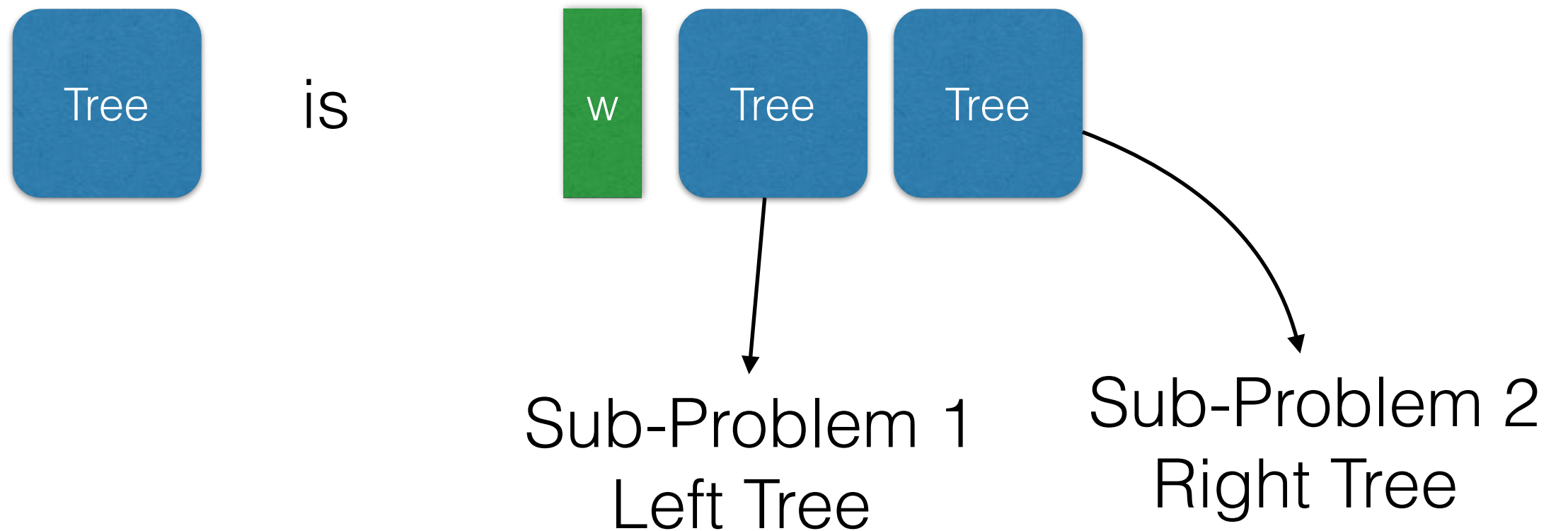# S-Expression

empty tree ::= ()

tree ::= empty tree | (w tree tree)

How to Parse

Tree **is** (w Tree Tree)

How to Parse

Tree is w Tree Tree

Sub-Problem 1
Left Tree

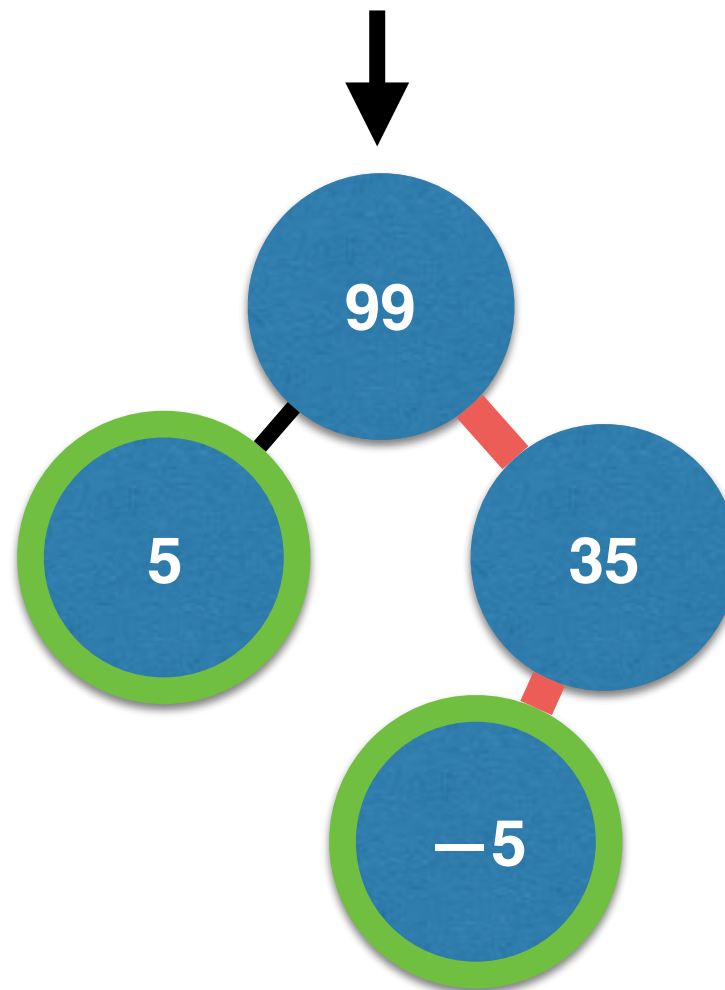Sub-Problem 2
Right Tree

Tree Structure

```cpp
class Node
{
public:
    int weight; // weight can be negative!
    Node *left, *right;
    Node():weight(0),left(NULL),right(NULL){}
    Node(int d):weight(d),left(NULL),right(NULL){}
};
```

Targets

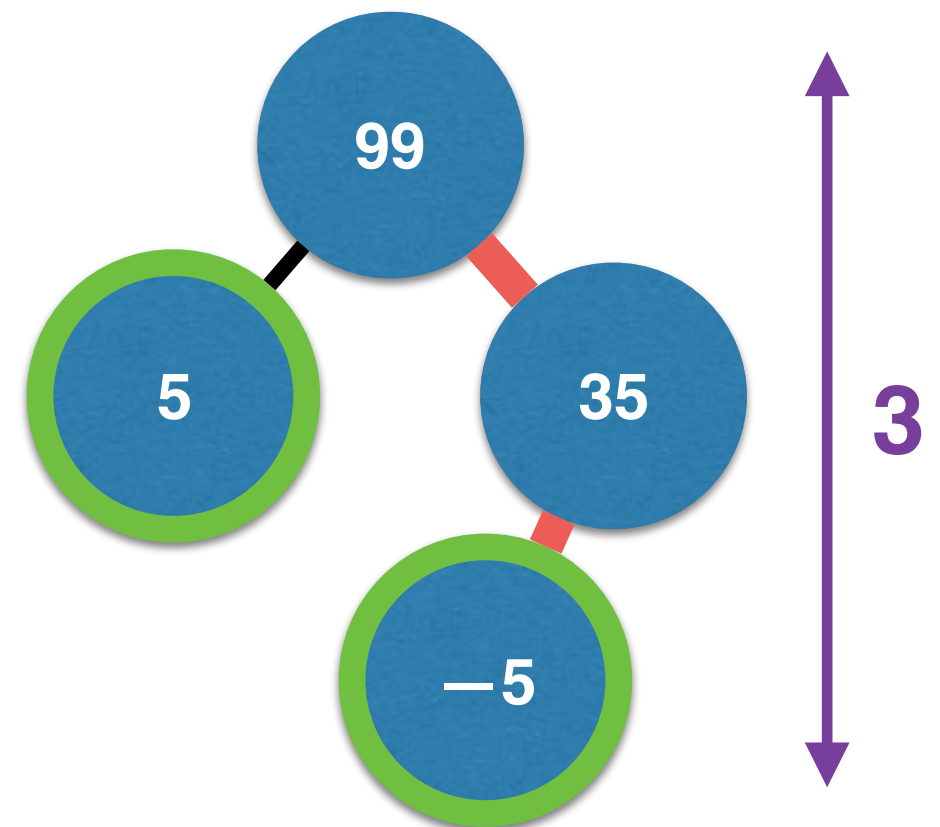(99(5()())(35(-5()())()))

constructTree

deleteTree

NULL

Targets

- treeHeight = 3

  - Height of the tree

- treeWeight = 134

  - Sum of node weights

- leafNum = 2

  - Leaf node number

- maxPathWeight = 129

  - Max weight from root to leaf

$$(99(5()())(35(-5()())()))$$

```cpp
//------------------------------------------------------------------
// Input a tree string in S-expression, construct the tree and
// return the pointer to its root node.
//------------------------------------------------------------------
virtual Node* constructTree(Node *root, std::string treeStr)const;


//------------------------------------------------------------------
// Delete and release the memory allocation of each node, and
// return a nullptr.
//------------------------------------------------------------------
virtual Node* deleteTree(Node *root)const;


//------------------------------------------------------------------
// Return the height of the tree.
//------------------------------------------------------------------
virtual int treeHeight(const Node *root)const;


//------------------------------------------------------------------
// Return the sum of node weights.
//------------------------------------------------------------------
virtual int treeWeight(const Node *root)const;


//------------------------------------------------------------------
// Return the number of leaf nodes.
//------------------------------------------------------------------
virtual int leafNum(const Node *root)const;


//------------------------------------------------------------------
// Return the max path weight from root to leaf.
//------------------------------------------------------------------
virtual int maxPathWeight(const Node *root)const;
```

Public Test Case

[S-Expression]
[treeHeight] [treeWeight] [leafNum] [maxPathWeight]

```
(99(5()())(35(-5()())()))¬
3 134 2 129¬
(99()(35(37()(14()(85()())))()))¬
5 270 1 270¬
(21()(94(76()(51()(25()()))))()))¬
5 267 1 267¬
(27(3()(-8()(30()(95()()))))(74(87(50()())(25(-1()(     .....
5 464 4 295¬
(68()(23()(-96(-16()())(63()()))))¬
4 42 2 58¬
(32()())¬
1 32 1 32¬
()¬
0 0 0 0
```

Hidden Test Cases

Warning!

- Each of the 3 hidden test cases contains up to **TEN-THOUSAND** trees!

- Each tree may have **HUNDREDS** of nodes.

Messages

- **[Undefined ExpressionEval::*******]**

  - The function is not implemented.

- **[Wrong ********]**

  - Different types of Wrong Answer

- **[Accepted]**

Submission

1. make clean

- remove object files (*.o) and the executable

2. Archive your source codes (whole hw3 folder) into a zip file named **[studentID]_hw3.zip**

- e.g. 104062999_hw3.zip

3. Submit the zip file to iLMS system BEFORE the deadline.