CSCI 241 Data Structures
Winter 2018
Assignment 2
Due, Wednesday, February 28

Submitting Your Work

This assignment is worth 15% of the grade for the course. Submit your Java program via Canvas in a zip folder titled `lastname_firstname_assignment2.zip`.

Your assignment will be evaluated on correct functionality and conformance to the coding standards described at the end of this assignment specification.

The Task

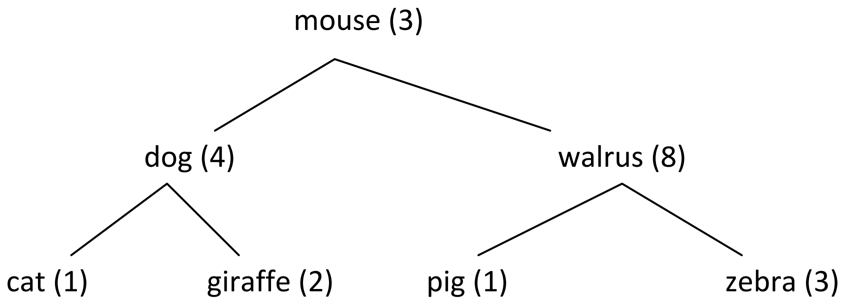
Your task is to modify the provided Java class BinarySearchTree, to ensure that the tree maintains AVL balance during insertions of new words into the tree. To facilitate this, you must add a height instance variable to the TreeNode class, representing the length of the longest path from a TreeNode to a descendent leaf TreeNode. Note that each leaf has height 1.

You must **NOT** modify anything in main() — BinarySearchTree acts as the driver class for this assignment and as such its functionality must remain consistent all across the board for grading. As seen in the BinarySearchTree Java class file, the insert() and dump() wrapper methods exist to call your own implementations of inserting and dumping methods. It is highly encouraged that you write additional methods within the BinarySearchTree class so as to not bloat your methods.

The dump() instance method for the BinarySearchTree class must display the full contents of the tree from a pre-order traversal of the tree. For each node in the tree, you must display, on a separate line, comma separated values for:

1. The word stored at that node.
2. The count of the word stored at that node.
3. The word stored in the node's parent node. Since the parent of the root node is null, display "*" instead.
4. The word stored at the left child of the node, or "*" if the node has no left child.
5. The word stored at the right child of the node, or "*" if the node has no right child.
6. The height of the node.

For example, given the following binary search tree, with word counts indicated in parentheses:



The output from dump() should be:

```
mouse, 3, *, dog, walrus, 3
dog, 4, mouse, cat, giraffe, 2
cat, 1, dog, *, *, 1
giraffe, 2, dog, *, *, 1
walrus, 8, mouse, pig, zebra, 2
pig, 1, walrus, *, *, 1
zebra, 3, walrus, *, *, 1
```

Coding Standards

1. Use meaningful names for variables that give the reader a clue as to the purpose of the thing being named.
2. Use comments at the start of the program to identify the purpose of the program, the author and the date written.
3. Use comments at the start of each method to describe the purpose of the method, the purpose of each parameter to the method, and the return value from the method (if any).
4. Use comments at the start of each section of the program to explain what that part of the program does.
5. Use consistent indentation.