Assignment 8

CS329E - Elements of Software Design
Binary Trees
(100 points)
Due Date on Canvas and Gradescope

1 Description

In this assignment you will be adding to the classes Node and Tree that we developed in Binary Search Tree Lecture and testing them. There are several short methods that you will have to write.

- Write a method *range()* that returns the range of values stored in a binary search tree of integers. The range of values equals the maximum value in the binary search tree minus the minimum value. If there is one value in the tree the range is 0. If the tree is empty the range is undefined.

  ```
  def range (self):
  ```

- Write a method *get_level()* that takes as input the level and returns a list of all the nodes at that level from left to right. If that level does not exist for that binary search tree return an empty list . Use the convention that the root is at level 0.

  ```
  def get_level (self, level):
  ```

- Write a method *left_side_view()* when given the root of a binary tree, imagine yourself standing on the left side of it, return the values of the nodes you can see ordered from top to bottom.

  ```
  def left_side_view (self):
  ```

- Write a method *sum_leaf_nodes()* that returns the sum of the value of all leaves. Recall that a leaf node does not have any children.

  ```
  def sum_leaf_node (self):
  ```

In this assignment you will be writing helper methods for the Tree class that we developed and test them. The following is the outline of the code that you will be submitting. You may include the other functions that we developed for completeness.

Input:

In the class TestBinaryTree you will create several trees and show convincingly that your methods are working. Here is an example of the file bst.in:

```
50 30 70 10 40 60 80 7 25 38 47 58 65 77 96
50 30 70 10 40 60 80 7 25 38 47 58 65 77 96
58 77 65 30 38 50 7 25 47 96 80 10 60 70 40
```

There should be enough documentation in your code that explains to the student assistants what you are testing and how. The file that you will be turning in will be called **TestBinaryTree.py**.

Pair Programming

For this assignment you may work with a partner. Both of you must read the paper on Pair Programming[1] and abide by the ground rules as stated in that paper. If you are working with a partner then only one of you will be submitting the code. But make sure that your partner's name and UT EID is in the header. If you are working alone then remove the partner's name and eid from the header.

1.1 Turnin

Turn in your assignment on time on Gradescope system on Canvas. For the due date of the assignments, please see the Gradescope and Canvas systems.

1.2 Academic Misconduct Regarding Programming

In a programming class like our class, there is sometimes a very fine line between "cheating" and acceptable and beneficial interaction between students (In different assignment groups). Thus, it is very important that you fully understand what is and what is not allowed in terms of collaboration with your classmates. We want to be 100% precise, so that there can be no confusion.

The rule on collaboration and communication with your classmates is very simple: you cannot transmit or receive code from or to anyone in the class in any way – visually (by showing someone your code), electronically (by emailing, posting, or otherwise sending someone your code), verbally (by reading code to someone) or in any other way we have not yet imagined. Any other collaboration is acceptable.

The rule on collaboration and communication with people who are not your classmates (or your TAs or instructor) is also very simple: it is not allowed in any way, period. This disallows (for example) posting any questions of any nature to programming forums such as **StackOverflow**. As far as going to the web and using Google, we will apply the **"two line rule"**. Go to any web page you like and do any search that you like. But you cannot take more than two lines of code from an external resource and actually include it in your assignment in any form. Note that changing variable names or otherwise transforming or obfuscating code you found on the web does not render the "two line rule" inapplicable. It is still a violation to obtain more than two lines of code from an external resource and turn it in, whatever you do to those two lines after you first obtain them.

Furthermore, you should cite your sources. Add a comment to your code that includes the URL(s) that you consulted when constructing your solution. This turns out to be very helpful when you're looking at something you wrote a while ago and you need to remind yourself what you were thinking.

We will use the following Code plagiarism Detection Software to automatically detect plagiarism.

- Staford MOSS

  `https://theory.stanford.edu/~aiken/moss/`

- Jplag - Detecting Software Plagiarism

  `https://github.com/jplag/jplag` and `https://jplag.ipd.kit.edu/`

[1] Read this paper about Pair Programming `https://collaboration.csc.ncsu.edu/laurie/Papers/Kindergarten.PDF`