### Binary Trees ( Due 20 Nov 2020 )

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

Write a method *is\_similar()* that takes as input two binary search trees and returns true if the nodes have the same key values and are arranged in the same order and false otherwise.

def is\_similar (self, pNode):

Write a method *print\_level()* that takes as input the *level* and prints out all the nodes at that level. If that level does not exist for that binary search tree it prints nothing. Use the convention that the root is at level 1.

def print\_level (self, level):

Write a method *get\_height()* that returns the height of a binary search tree. Recall that the height of a tree is the longest path length from the root to a leaf.

def get\_height (self):

Write a method *num\_nodes()* that returns the number of nodes in the left subtree from the root and the number of nodes in the right subtree from the root and the root itself. This function will be useful to determine if the tree is balanced.

def num\_nodes (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.

import sys

class Node (object):

...

class Tree (object):

# Returns true if two binary trees are similar

def is\_similar (self, pNode):

# Prints out all nodes at the given level

def print\_level (self, level):

# Returns the height of the tree

def get\_height (self):

# Returns the number of nodes in the left subtree and

# the number of nodes in the right subtree and the root

def num\_nodes (self):

def main():

# Create three trees - two are the same and the third is different

line = sys.stdin.readline()

line = line.strip()

line = line.split()

tree1\_input = list (map (int, line)) # converts elements into ints

line = sys.stdin.readline()

line = line.strip()

line = line.split()

tree2\_input = list (map (int, line)) # converts elements into ints

line = sys.stdin.readline()

line = line.strip()

line = line.split()

tree3\_input = list (map (int, line)) # converts elements into ints

# Test your method is\_similar()

# Print the various levels of two of the trees that are different

# Get the height of the two trees that are different

# Get the total number of nodes a binary search tree

if \_\_name\_\_ == "\_\_main\_\_":

main()

To run this code on the Mac you will do:

python3 TestBinaryTree.py < bst.in

On a Windows machine you will do

python TestBinaryTree.py < bst.in

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](https://www.cs.utexas.edu/users/mitra/csFall2020/cs313/assgn/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**. The file will have a header of the following form:

# File: TestBinaryTree.py

# Description:

# Student Name:

# Student UT EID:

# Partner Name:

# Partner UT EID:

# Course Name: CS 313E

# Unique Number:

# Date Created:

# Date Last Modified:

For this assignment you may work with a partner. If you do, both of you must read the paper on [Pair Programming](https://www.cs.utexas.edu/users/mitra/csFall2020/cs313/assgn/PairProg-CACM-1999.pdf)and submit only **one** copy of the program but make sure that you have your partner's name and eid in your program. If you are doing this assignment by yourself, then remove the *Partner Name* and *Partner UT EID* from the header.

Use the [Canvas](http://canvas.utexas.edu/)system to submit your **TestBinaryTree.py** file. We should receive your work by 11 PM on Friday, 20 Nov 2020. There will be substantial penalties if you do not adhere to the guidelines. Remember Python is case sensitive. The name of your file must match exactly what we have specified.

* Your Python program should have the proper header.
* Your code must run before submission on the command line.
* You should be submitting your file through the web based *Canvas* program. We will not accept files e-mailed to us.