CSC148 Summer 2018: Exercise 7

Due: Thursday, July 19th @ 11PM

In this exercise, you are to implement a function called get_largest_height_difference().

To start, download ex7.py and ex7_pyta.txt and read through the code provided in the `if __name__ == '__main__'` block.

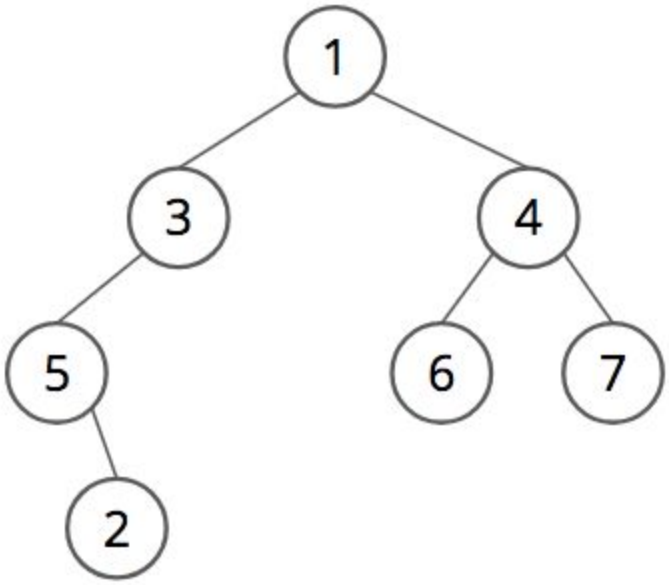
Read through the docstring for get_largest_height_difference() carefully and implement the method. We have provided the ___init___, and ___str___ methods for convenience, as well as the get_height() function.

This exercise will require you to have PythonTA installed. If you haven't done so already, go through lab1 and the instructions on the Software page to install and set up PyCharm with PythonTA.

get_largest_height_difference

This method takes in a BinaryTree and returns the largest difference in height between 2 subtrees of a BinaryTree.

For example, consider the BinaryTree t:



The height difference between t's left and right subtrees is 1.

The height difference of t.right is 0, since both children have the same height.

The height difference of t.left is 2, since its left subtree has a height of 2 while the right has a height of 0.

Thus, the maximum height difference returned by t is 2.

Submission

Exercises are to be submitted through MarkUs in the ex6 folder. Submit only ex6.py.

To log in to MarkUs, use your UTORid as the log-in name. The password is your teaching labs password. If you have not set this up or have forgotten your password, go to the Teaching Lab's Account Management Page and (re)set your password.

Grading Scheme

This exercise will be graded out of 4 marks, broken down as follows:

- 2 marks for being able to run the client code without issue (no assertion errors raised)
- 1 mark for passing PythonTA
- 1 mark for passing hidden test cases (which use your client code in other ways)
    - Details on what the hidden test cases will/won't test are describe below.

All of these marks are 'all-or-nothing' (i.e. you'll either get 0 on that criteria, or full marks).

Hidden Test Cases

Things that the hidden test case might test:

- Any BinaryTree imaginable.

(So any combination of different heights of subtrees -- any BinaryTree you can draw on paper, we can test.)