Kerry Ojakian's CSI 33 Class
Due Date: Tuesday November 19

HW #4

General Instructions: Problems 1 to 6 (and the written part of problem 7) should be handed in on paper at the beginnning of class. For problem 7, create a folder HW04 in your folder Homework at Dropbox. Put one Juypter file in HW04 for problem 7 (yes, you will lose points for not following instructions ...).

The Assignment

Except for the last question, the problems are all written problems. Include all SIX written questions, in order, in ONE .pdf file. For the last question (Problem 7), create a single Jupyter notebook file.

1. (a) Draw a binary tree with 6 nodes and 3 leaves.
   (b) Draw a binary tree with 6 nodes and 2 leaves.
   (c) Draw a complete binary tree with 9 nodes. How many leaves does it have?

2. Short answer question 6 from page 248 of chapter 7. Note: the algorithm referred to means to do the only reasonable thing - compute from the bottom of the tree, up to root.

3. Short answer question 7 from page 248 of chapter 7 (it refers to the tree on **page 231**). Do this problem in TWO ways using our TreeNode and BinaryTree classes. Note: Just write your Python code by hand (or typed) in the single PDF with the rest of the written work.

   (a) Do it from scratch, just using the TreeNode class, without using the BinaryTree class (i.e as asked for in the book problem).
   (b) Do it using the addNode method from BinaryTree class that we wrote (beyond what is asked for in the book). Remember, that has addresses of nodes.

4. Consider the tree on **page 231** again. Write down the ordering you get for each of the following kinds of traversals: Inorder, Postorder, and Preorder.

5. Find a prefix code for the 5 symbols: A, I, M, T, SPACE. It should save on the brute force approach of using 3 bits per symbol; save as much as possible Write the phrase ”I AM IT” using your code.

6. Calculate the following. Also for each expression, draw a corresponding expression tree, and use that tree to convert each to in-order notation.

   (a) postfix expression: $1\ 3\ -\ 4\ 2\ 1\ /\ *\ +$

   (b) prefix expression: $-\ +\ 4\ 3\ /\ 4\ 8$

7. (**Python Problem**) Create a guessing game which will ask the user a series of yes/no questions, and after the series of questions, gives its guess to what the user is thinking of. To do this, you should pick some category of objects and inform the user what this category is (for example: Animals; you may **not** use Animals-pick a different category!). Then the program asks a series of yes/no questions, and after some number of questions, it guesses the object (for example: Dog). As an example of the interaction between the user and computer see the book, pages 251-252, starting at “*Welcome to the Animal Game!*” and ending on the next page on line 11 at “*Rats! I didn’t get it.*”

   Note: You are **not** doing problem 10 on page 251, which is significantly harder. All you should do is hardcode one particular decision tree which is used, unmodified, each time the program is run. To say that again: Your program should build this one particular decision tree, and then the core operation of your program will be to start at the root and navigate to the leaf according to the user responses. See our animal guessing game from class.

   Also, draw a clear picture of your decision tree to hand in with problems 1 to 6.