COMP 221-01 Homework 3

Due: Friday, October 4, by 11:59 pm

1 Guidelines

Please **type up** and submit a PDF of your solutions. If you have to draw any pictures for a question (such as trees, graphs, stacks, or queues operations), you can do so **neatly** and then add a picture of it to your PDF file.

If a problem is explicitly marked as a **Programming** question, you need to submit the source code for it (unless otherwise indicated).

Programming language: You are welcome to use Python or Java.

Do your own work: Homework should be individual work. You may discuss problems with other people, but you must write up your solutions yourself. I will not say that the web is off limits, but handing in solutions you find online as if they are your own is also not acceptable. Speaking of, most of these questions are shamelessly stolen from Susan :)

Make sure you have your name on your homework!

- **Question 1**

  From Skiena (slightly modified), Question 3-1 (page 98).

  A common problem for compilers and text editors is determining whether the parentheses (or braces or square brackets) in a string are balanced and properly nested. For example, the string [(({()})())()] contains properly nested pairs of parentheses, while the strings)(){(and ())[do not. Create an algorithm (in **pseudocode**) that returns TRUE if a string contains properly nested and balanced parentheses, and FALSE if otherwise. Hint: What data structure (that we've reviewed) would make this task a lot easier?

- **Question 2 - BSTs**

  1. Draw the BST that would result from inserting the following values in the order shown.

     41 25 83 99 15 5 1 34 96 55

  2. Give the pre-order traversal for this BST.
  3. Give the in-order traversal for this BST.
  4. Give the post-order traversal for this BST.

- **Question 3 - Programming**

  I am providing you with Susan's starter implementation of a binary search tree (using classes, in both Java and Python), without the **insert** method. Your task is to understand the starter code (feel free to modify it however you like), and then to implement and test the **insert** method.

  You should not use code from any other source. Submit your code separate from your homework document. However, in your homework write-up, summarize the main methods of the BST code, and describe how you implemented and tested the insert method. **Please copy your insert method code into your homework write-up.** The preceptors will only be grading the method as written in your homework, and will not run your code unless they notice a discrepancy.

  Note that test cases have been written for you - if you feel like you need more, you are welcome to add them!

- **Question 4 - Programming**

  Complete the "I Can Guess the Data Structure" problem on Open Kattis:

  https://open.kattis.com/problems/guessthedatastructure

  Try to pass all the tests if you can (with Kattis it's hard because they don't give you all the test cases). If you can't pass all of them, give a brief write-up for why you think you didn't pass.

  Please include a screenshot of your Kattis test passed page in your homework write-up.