**CSC 310 Data Structures**

**Assignment 4**

**Total: 50 points**

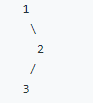
Due: Thursday, October 31 at 11:59 PM

**Project Requirements**

1. **(15 points)** Write an insertion-sort program using a priority queue implemented with a sorted list. Note that follow a Priority Queues lecture slide pp. 12-13.
2. **(20 points)** Implement a program which returns i) inorder and ii) preorder traversal of a given a binary tree. Note that you should follow the code fragment style to create your program below and can use recursive (or iterative) function.

Example:

Input: [1, null, 2, 3]



Output:

inorder: [1, 3, 2]

preorder: [1, 2, 3]

# Definition for a binary tree node.

class TreeNode(object):

def \_\_init\_\_(self, x):

self.val = x

self.left = None

self.right = None

class YourSolution(object):

def inorderTraversal (self, root):

# :type root: TreeNode

# :rtype: List[int]

def preorderTraversal(self, root):

#:type root: TreeNode

#:rtype: List[int]

1. **(15 points)** Create a private GitHub repo to post your programming sources codes files of the homework 4, README file, data (if applicable), and screenshots. Send me a GitHub URL with a password. Note that this is an official GitHub tutorial for the beginners and a good example for the GitHub repo.

**•** GitHub tutorial: https://guides.github.com/activities/hello-world/

• Michael Gallier’s GitHub: https://github.com/galliersm/CSC-310-Project-1

**Programming Requirements**

You are required to use **Python (or JAVA)** for this homework.

**Submission Requirements (Still, you need to upload the submission deliverable to Blackbaord)**

Your homework submission must be done **originally**. Please submit **TWO** files, separately:

1. **Single tar (or zip) file** that include **FOUR** files below:
   * **Source codes**: your Python or (JAVA) codes and a shell script files (if applicable) that you use for testing. For the full credits, your code also needs to be well-documented, with any major constructs (i.e., functions) clearly commented.
   * **README:** A Text file (i.e., .TXT), includingoverall high-level documentation of instructions to run your programs.
   * **Screenshot files:** Screenshot files should be readable to gain the full credits. The instructor will use the screenshots to compare with a student’s actual program testing.
   * **Data**: input files to be tested if applicable.
2. **MS word (or Acrobat PDF) file**, including ONLY source codes for originality checking.

**Late & Plagiarism Policy**

If you use any programming design (or approach) from the references, please add the citation to the Report. Please check the late/plagiarism policy on the course syllabus.