Computer Science

COMPSCI 105 S2 C - Assignment 2

Part 1 due on Friday 12th October 2018 at 6:00pm
Part 2 due on Friday 19th October 2018 at 6:00pm

Assessment

- Due: Part 1: **6:00pm, 12th of October 2018** and Part 2: **6:00pm, 19th October 2018**
- Worth: 100 marks in total = 5% of the final grade

Warning

DO NOT SUBMIT SOMEONE ELSE'S WORK:

- The work done on this assignment must be your own work. Think carefully about any problems you come across, and try to solve them yourself before you ask anyone for help
- Under no circumstances should you take or pay for a copy of someone else's work. This will be penalized heavily.
- Under no circumstances should you give a copy of your work to someone else
- The Computer Science department uses copy detection tools on the files you submit. If you copy from someone else, or allow someone else to copy from you, this copying will be detected and disciplinary action will be taken

To ensure you are not identified as cheating you should:

- Always do individual assignments by yourself
- Never give another person your code
- Never put your code in a public place (e.g. forum or website)
- Never leave your computer unattended, you are responsible for the security of your account
- Ensure you always remove your USB flash drive from the computer before you log off

Part II of Assignment

Resources and Submission

- All submitted files must contain **your name and UPI in a comment at the top of the file**.
- All your files should be able to be compiled without requiring any editing.
- All your files should include a good layout structure, meaningful variable names, and comments explaining the key ideas of your solution
- All required resources are found on the assignment section in the course web page

Q4: Resources –`ListBinaryTree.py, ReconstructTree.py`
Submission –`ReconstructTree.py`

Aims of Part II of Assignment

Understanding and solving problems using:

- Binary trees

Q4. Constructing a Binary Tree *(40 Marks)*

In this exercise we develop a method to reconstruct a binary tree from an inorder and postorder traversal sequence of the unknown tree, as well as provide the postorder traversal sequence of the tree.

Please complete the program `ReconstructTree.py`, which lets the user enter the inorder and postorder traversal sequences of a tree as input arguments into the function `buildTree()`. The function then constructs the corresponding binary tree. The program also needs to include the function `postorder()`, which returns the postorder traversal sequence of an input tree.

Please note:

- All nodes of the tree contain a single character and the inorder and postorder traversal sequences are strings formed by these characters in the corresponding order.
- Note that in our examples the reconstructed tree is a binary tree, but not a binary search tree
- Please construct the tree using a list-of-list representation and use the supplied file `ListBinaryTree.py`
- All of your own implementations should be contained within `ReconstructTree.py`

The following figures show test cases and expected output of the program.

```
Binary Tree construction:
Please enter the inorder sequence: 42513
Please enter the preorder sequence: 12453
[1, [2, [4, None, None], [5, None, None]], [3, None, None]]
Postorder traversal of tree is: 45231
```

```
Binary Tree construction:
Please enter the inorder sequence: CS105
Please enter the preorder sequence: 0SC15
[0, [S, [C, None, None], [1, None, None]], [5, None, None]]
Postorder traversal of tree is: C1S50
```

```
Binary Tree construction:
Binary Tree construction:
Please enter the inorder sequence: cbdae
Please enter the preorder sequence: abcde
[a, [b, [c, None, None], [d, None, None]], [e, None, None]]
Postorder traversal of tree is: cdbea
```

Marking Scheme for Question 4

Correct root of tree	5 marks
Correct level 1 of tree	10 marks
All higher levels of tree are correct	15 marks
Correct implementation of the `postorder()` function	10 marks