Lecture Notes – Trees

Overview

Lists, stacks, queues are all linear data structures.

Trees are 2-dimensional, hierarchical structures.

Refer to the e-text section 7.3 for some terminologies, concepts and definitions.

https://runestone.academy/runestone/books/published/pythonds/index.html

General tree vs. Binary tree

A general tree is a tree in which each node can have an unlimited number of children nodes. Although general trees have less use in computing applications, the concept is common in many applications.
A binary tree is a tree in which each node has at most 2 children nodes, a left subtree and a right subtree.

```
class Node:
        def __init__ (self, data):
                self.data = data
                self.left = None
                self.right = None
```

to create a node holding an integer 10:
newNode = Node(10)

In the e-text, this concept of the Node is named BinaryTree (section 7.5):

```
class BinaryTree:
  def __init__(self,rootObj):
    self.key = rootObj
    self.leftChild = None
    self.rightChild = None

  def insertLeft(self,newNode):
    if self.leftChild == None:
      self.leftChild = BinaryTree(newNode)
    else:
      t = BinaryTree(newNode)
      t.leftChild = self.leftChild
      self.leftChild = t
```

```
    def insertRight(self,newNode):
        if self.rightChild == None:
            self.rightChild = BinaryTree(newNode)
        else:
            t = BinaryTree(newNode)
            t.rightChild = self.rightChild
            self.rightChild = t


    def getRightChild(self):
        return self.rightChild

    def getLeftChild(self):
        return self.leftChild

    def setRootVal(self,obj):
        self.key = obj

    def getRootVal(self):
        return self.key


r = BinaryTree('a')
print(r.getRootVal())
print(r.getLeftChild())
r.insertLeft('b')
print(r.getLeftChild())
print(r.getLeftChild().getRootVal())
r.insertRight('c')
print(r.getRightChild())
print(r.getRightChild().getRootVal())
r.getRightChild().setRootVal('hello')
print(r.getRightChild().getRootVal())
```