

## CS 5012: Binary Trees Module Exercise

### Question 1:

Assume the implementation of a `BinaryTree` from slide 30 in *Introduction to Trees*. Draw the binary tree that the following code produces (the code was written in *main*).

```
# create a bunch of nodes with integer values

theRoot = BinaryTree.Node(3) # root node with value 3

n1 = BinaryTree.Node(1)
n2 = BinaryTree.Node(2)
n4 = BinaryTree.Node(4)
n5 = BinaryTree.Node(5)
n6 = BinaryTree.Node(6)

# create a binary tree called 'myTree'

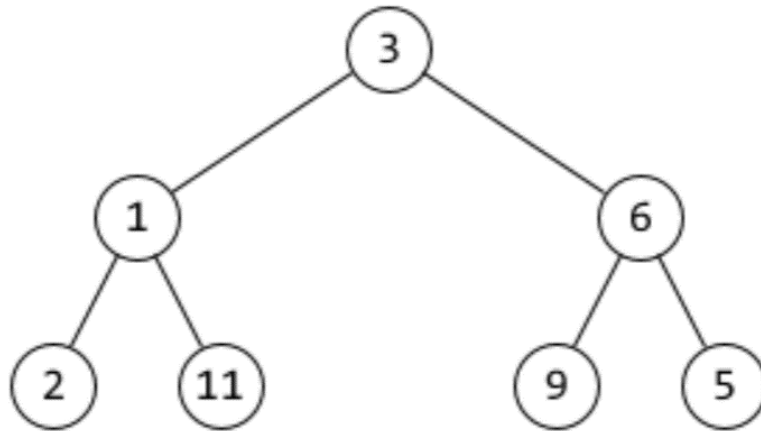
myTree = BinaryTree(theRoot) # create a tree 'myTree' with root = 3

# connect the tree

myTree.root.setLeft(n1)
myTree.root.setRight(n4)
n1.setRight(n2)
n4.setRight(n6)
n4.setLeft(n5)
```

### Question 2:

Assume the implementation of a `BinaryTree` from slide 30 in *Introduction to Trees*. Build a binary tree by providing the code in Python as would be typed in main (similar to the code in Q1), based on the following drawing.



### Question 3:

Write a tree-level `getHeight()` method that calculates the height (or depth) of the binary tree (e.g., it would return 3, if called on the tree in Q2).

Note: Think of a way to do this recursively, having each node calculate its height, as if it was the root of its subtree.