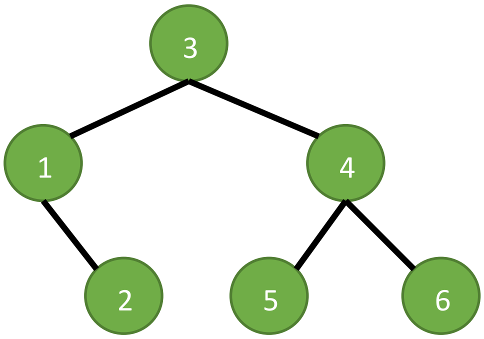
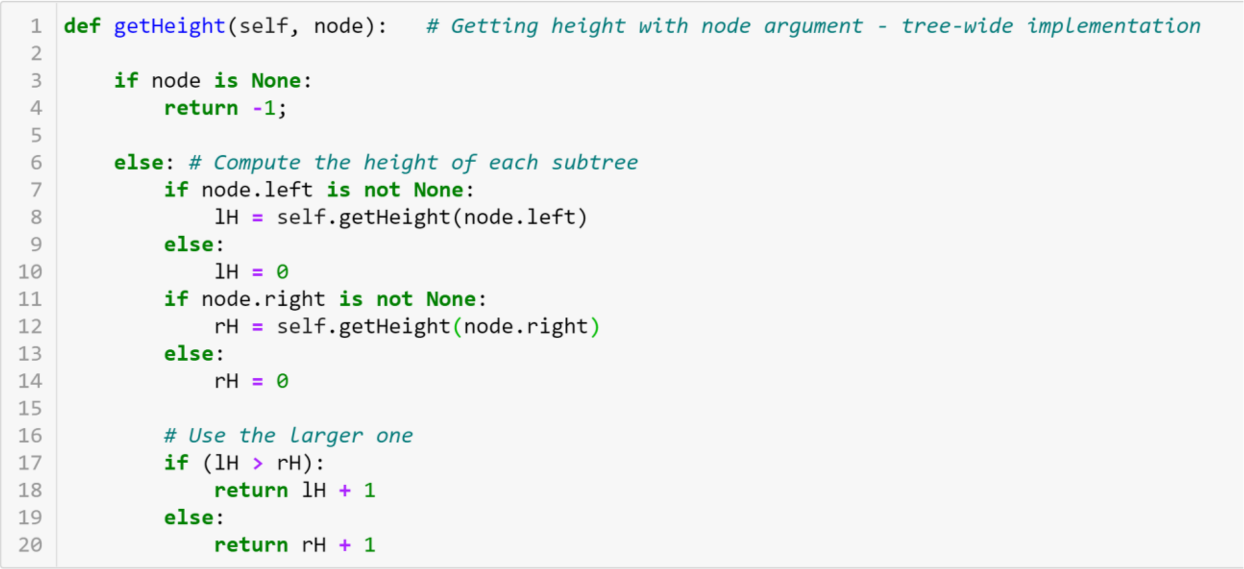
Q1.



Q2.

```
theRoot = BinaryTree.Node(3)

n1 = BinaryTree.Node(1)

n2 = BinaryTree.Node(2)

n5 = BinaryTree.Node(5)

n6 = BinaryTree.Node(6)

n9 = BinaryTree.Node(9)

n11 = BinaryTree.Node(11)

myTree = BinaryTree(theRoot)

myTree.root.setLeft(n1)

myTree.root.setRight(n6)

n1.setLeft(n2)

n1.setRight(n11)

n6.setLeft(n9)

n6.setRight(n5)
```

Q3.



```python
def getHeight(self, node):    # Getting height with node argument - tree-wide implementation

    if node is None:
        return -1;

    else: # Compute the height of each subtree
        if node.left is not None:
            lH = self.getHeight(node.left)
        else:
            lH = 0
        if node.right is not None:
            rH = self.getHeight(node.right)
        else:
            rH = 0

        # Use the larger one
        if (lH > rH):
            return lH + 1
        else:
            return rH + 1
```