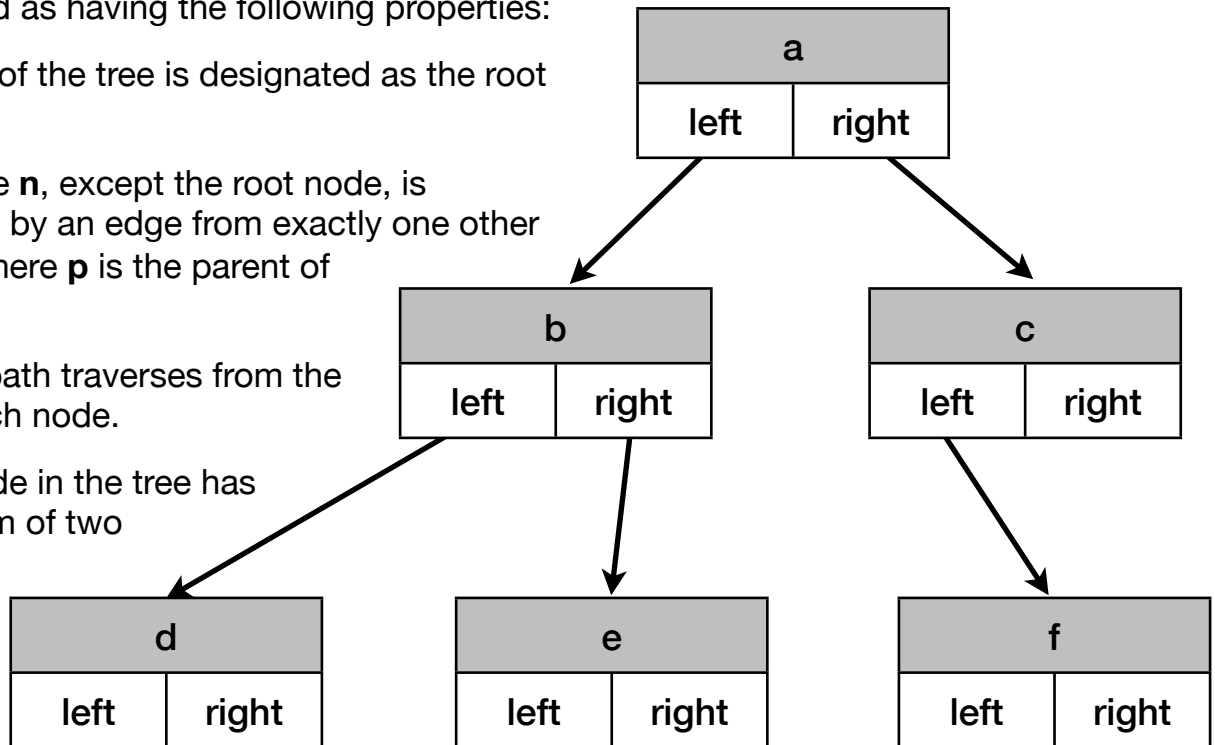


Trees

A tree is defined as having the following properties:

- One node of the tree is designated as the root node.
- Every node **n**, except the root node, is connected by an edge from exactly one other node **p**, where **p** is the parent of **n**.
- A unique path traverses from the root to each node.
- If each node in the tree has a maximum of two children, we say that the tree is a **binary tree**.



BinaryTree
root
left
right
getLeftChild
getRightChild
setRootVal
getRootVal
insertLeft
insertRight

getLeftChild() - returns the binary tree corresponding to the left child of the current node.
getRightChild() - returns the binary tree corresponding to the right child of the current node.
setRootVal(val) - stores the object in parameter val in the current node.
getRootVal() - returns the object stored in the current node.
insertLeft(val) - creates a new binary tree and installs it as the left child of the current node. If an existing tree is present the existing tree becomes the left node of the new tree.
insertRight(val) - creates a new binary tree and installs it as the right child of the current node. If an existing tree is present the existing tree becomes the right node of the new tree.

Implementation

Use the above class definition and method descriptions to help you produce a simple implementation of a binary tree. You may want to investigate the **`__str__()`** **method** that is available in Python to generate more useful messages when printing out objects.