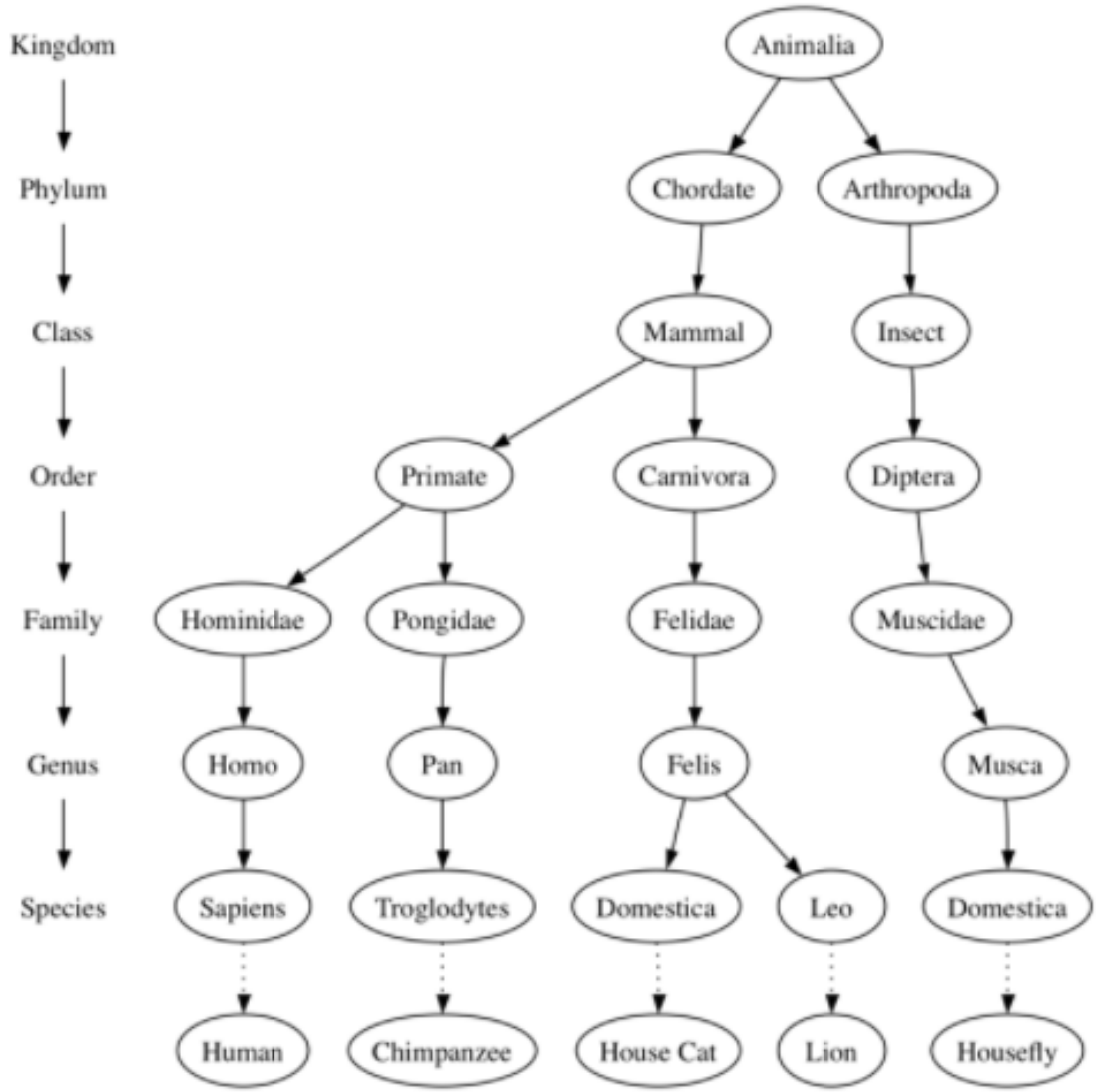
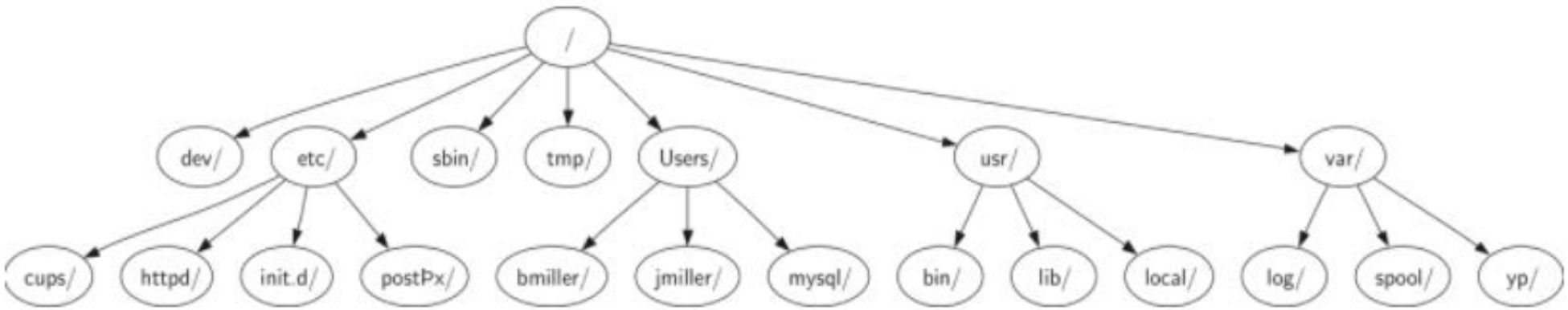
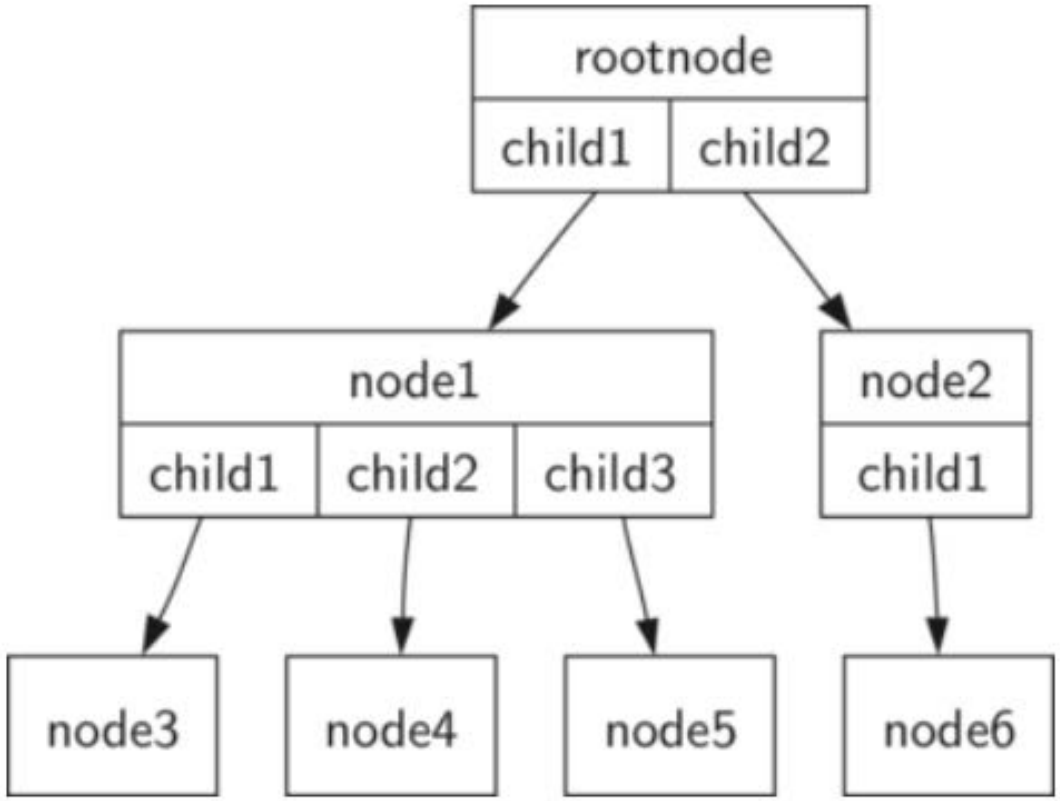

Kingdom
Phylum
Class
Order
Family
Genus
Species
Animalia
Chordate
Arthropoda
Mammal
Insect
Primate
Carnivora
Diptera
Hominidae
Pongidae
Felidae
Muscidae
Homo
Pan
Felis
Musca
Sapiens
Troglodytes
Domestica
Leo
Domestica
Human
Chimpanzee
House Cat
Lion
Housefly


/
dev/
etc/
sbin/
tmp/
Users/
usr/
var/
cups/
httpd/
init.d/
postPx/
bmiller/
jmiller/
mysql/
bin/
lib/
local/
log/
spool/
yp/



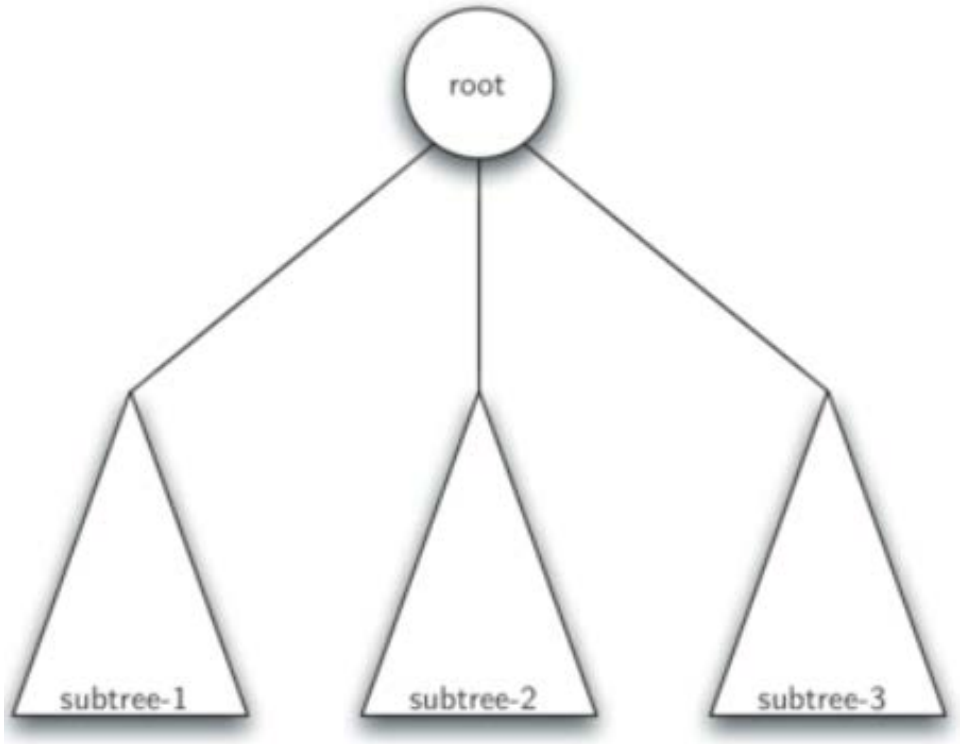
A tree consists of a set of nodes and a set of edges that connect pairs of nodes. A tree has the following properties:

•One node of the tree is designated as the root node.

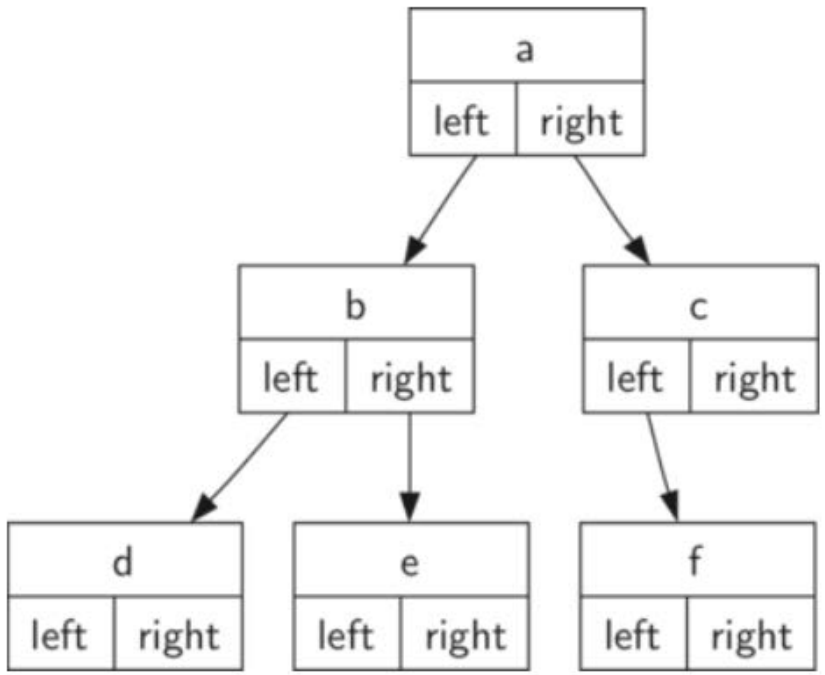
•Every node n, except the root node, is connected by an edge from exactly one other node p, where p is the parent of n.

•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**.

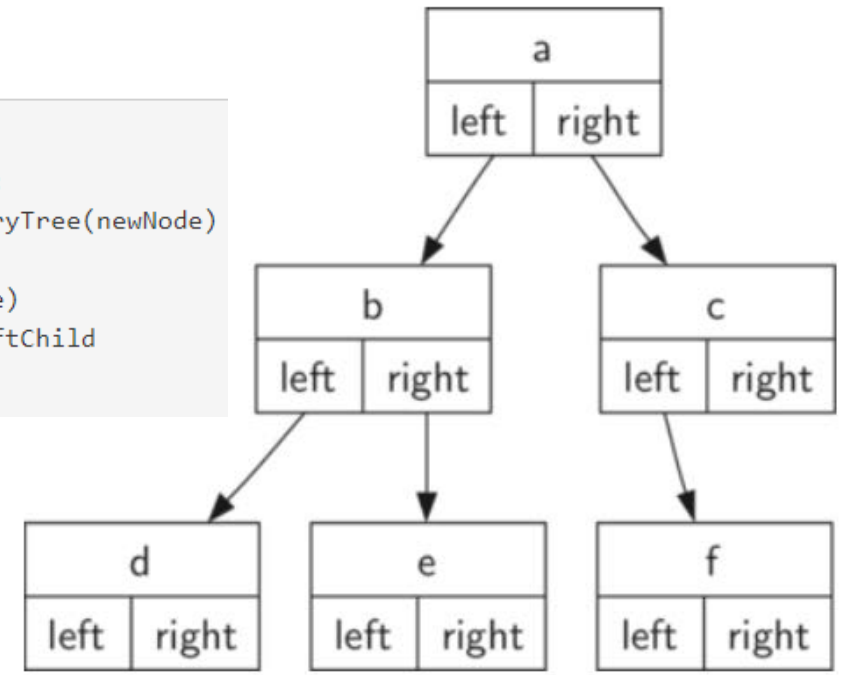


A tree is either empty or consists of a root and zero or more subtrees, each of which is also a tree. The root of each subtree is connected to the root of the parent tree by an edge. This figure illustrates this recursive definition of a tree. Using the recursive definition of a tree, we know that this tree has at least four nodes, since each of the triangles representing a subtree must have a root.




```
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
```