



数据结构与算法 (Python版)

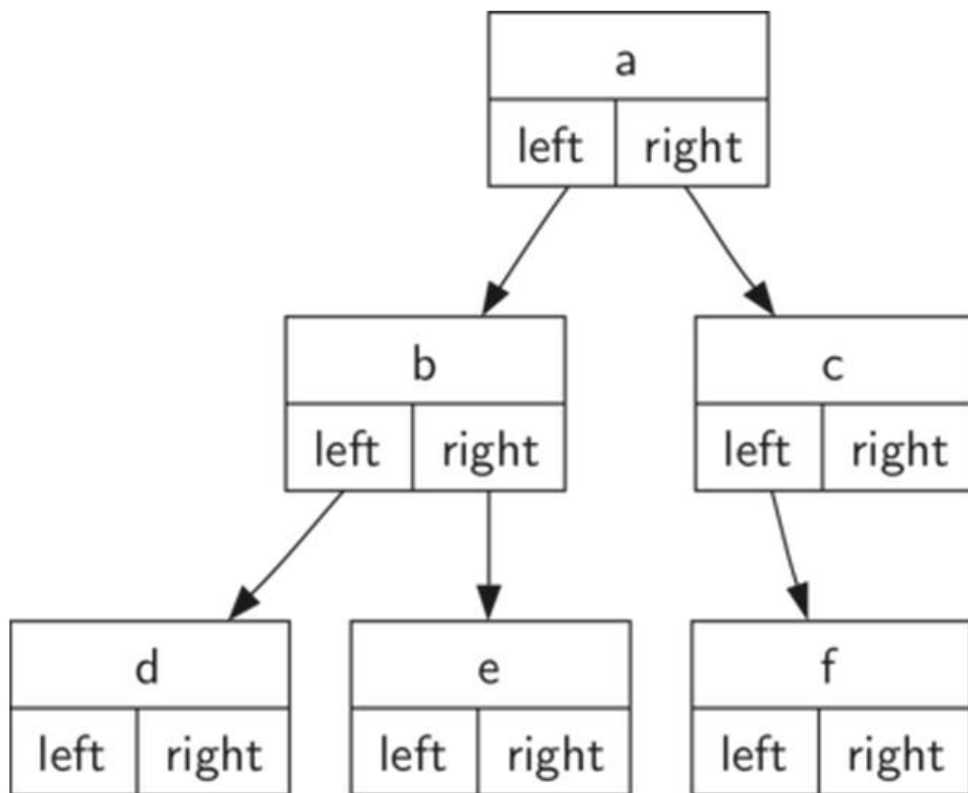
树的链表实现

陈斌 北京大学 gischen@pku.edu.cn

实现树：节点链接法

❖ 同样可以用节点链接法来实现树

每个节点保存根节点的数据项，以及指向左右子树的链接



实现树：节点链接法

❖ 定义一个BinaryTree类

成员key保存根节点数据项

成员left/rightChild则保存指向左/右子树的引用（同样是BinaryTree对象）

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

实现树：节点链接法

```
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
```

实现树：节点链接法

```
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
```

实现树：节点链接法

❖ 请画出r的图示

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

