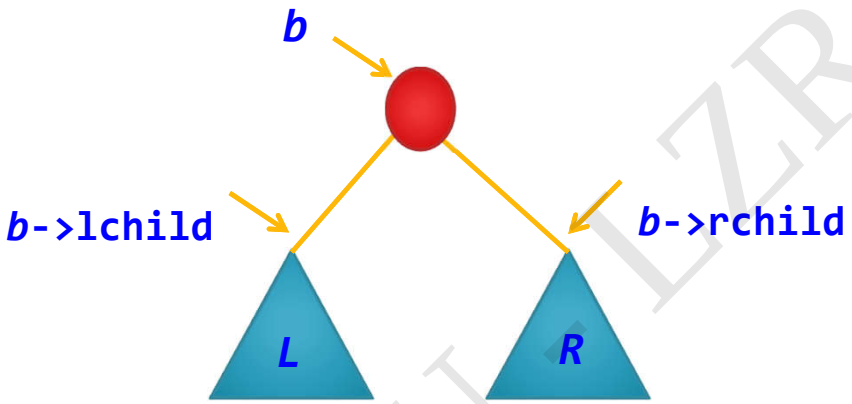
二叉树递归算法设计示例

【示例-1】设计求二叉树叶子结点个数算法。



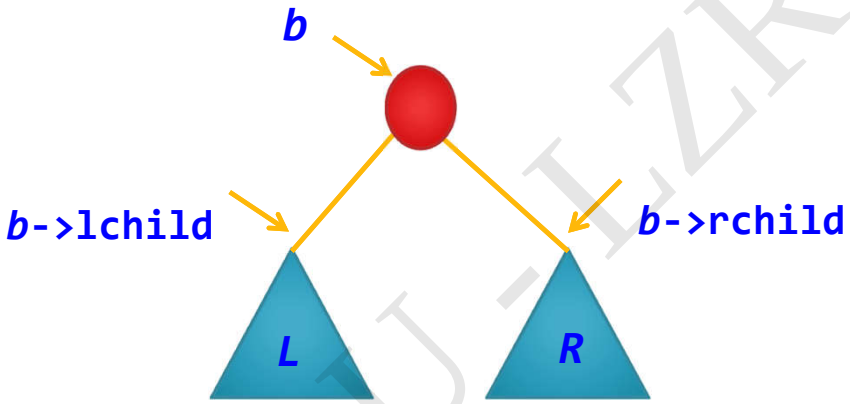
求二叉树b的叶子结点个数的递归模型f(b)如下：

$f(b)=0$	当b=NULL
$f(b)=1$	当b为叶子结点
$f(b)=f(b\text{->}lchild)+f(b\text{->}rchild)$	其他情况

求二叉树T的叶子结点个数的递归代码如下：

```
int LeafCount(BTNode *T)      //求二叉树T的叶子结点个数
{
  int lnum, rnum;
  if (T == NULL)              //空树返回0
    return 0;
  else if (T->lchild == NULL && T->rchild == NULL)
    return 1;          //为叶子结点时返回1
  else {
    lnum = LeafCount(T->lchild);     //求左子树叶子结点个数
    rnum = LeafCount(T->rchild);     //求右子树叶子结点个数
    return (lnum + rnum);            //返回和
  }
}
```

【示例-2】设计算法交换二叉树所有分支的左右孩子。



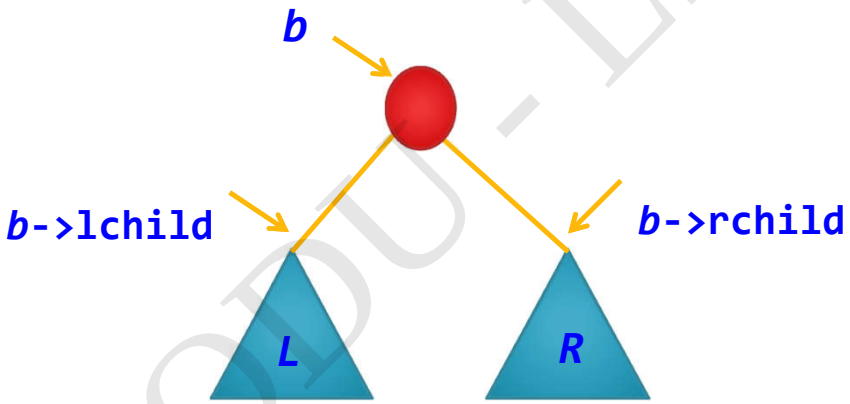
交换二叉树b所有分支的左右孩子的递归模型f(b)如下：

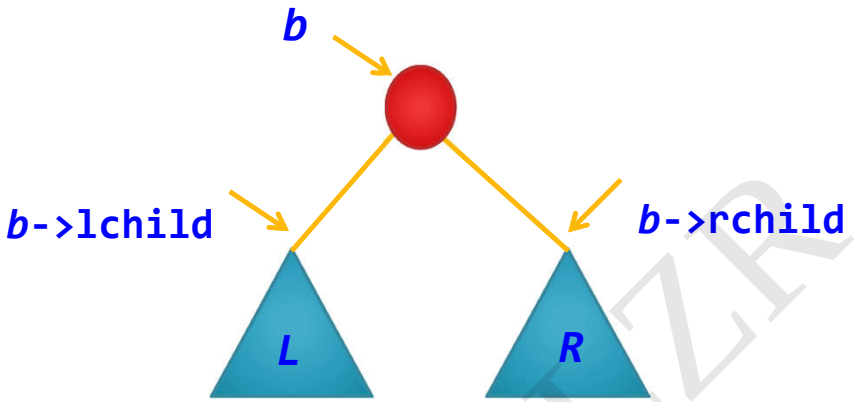
f(b)=什么都不做	当b=NULL
f(b)=什么都不做	当b为叶子结点
f(b)= 交换指针 b->lchild和b->rchild	其他情况

交换二叉树T的孩子结点的递归代码如下：

```
int Exchange(BiTNode *T)
{
    if(T != NULL)
        if(T->lchild != NULL || T->rchild != NULL)) {
            BiTNode *s = T->lchild;
            T->lchild = T->rchild;
            T->rchild = s;              //交换*t的两个子女
            Exchange(T->lchild);        //交换左子树的子女
            Exchange(T->rchild);        //交换右子树的子女
        }
}
```

【示例-3】假设二叉树中每个结点值为单个字符，采用二叉链表存储结构存储。设计一个算法计算一棵给定二叉树 b 中的所有单分支结点个数。





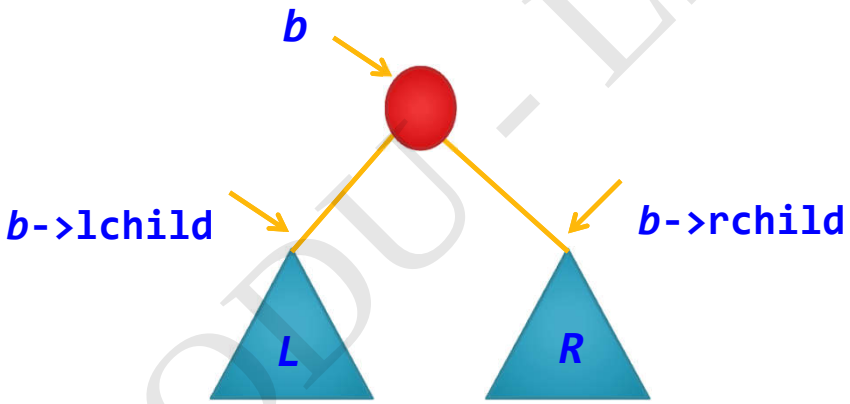
计算一棵给定二叉树 b 中的所有单分支结点个数的递归模型f(b)如下：

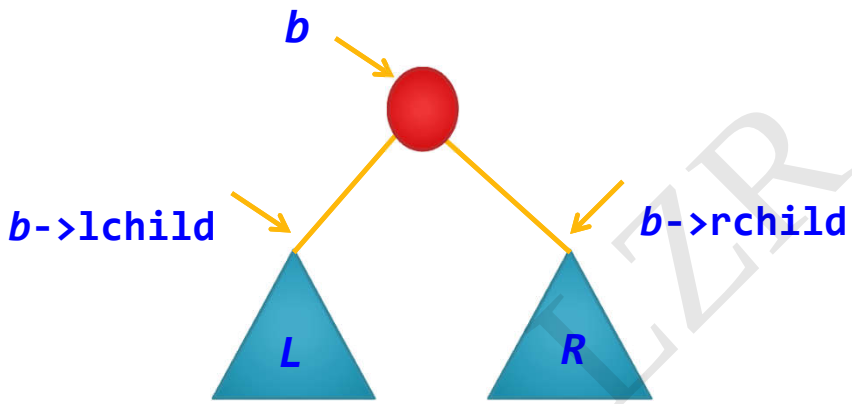
f(b)=0	b=NULL
f(b)=f(b->lchild)+f(b->rchild)+1	若b结点为单分支
f(b)=f(b->lchild)+f(b->rchild)	其他情况

计算所有单分支结点个数的递归代码如下：

```
int DegreeOne(BiTNode *b)
{
    int lnum, rnum, n;
    if(b == NULL)
        return 0;
    else
        if((b->lchild == NULL && b->rchild != NULL) ||
                (b->lchild != NULL && b->rchild == NULL))
            n = 1; //为单分支结点
        else
            n = 0; //其他结点
    lnum = DegreeOne(b->lchild); //递归求左子树中单分支结点数
    rnum = DegreeOne(b->rchild); //递归求右子树中单分支结点数
    return (lnum + rnum + n);
}
```

【示例-4】假设二叉树中每个结点值为单个字符，采用二叉链表存储结构存储。设计一个算法求二叉树b中最小值的结点值。





求二叉树b中最小值的递归模型f(b)如下：

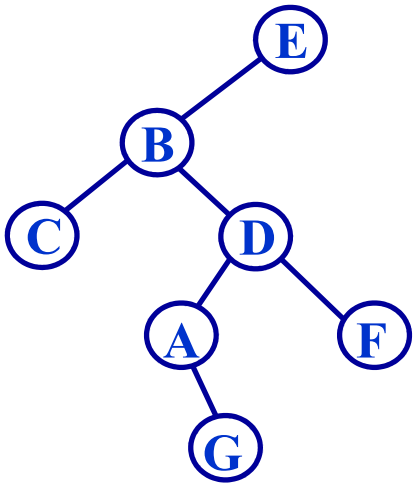
```
f(b, min)=不做任何事件                                          b=NULL
f(b, min)=当 b->data<min 时置 min=b->data;
      f(b->lchild,  min); f(b->rchild,  min);  其他情况
```

求二叉树 b 中最小值的递归代码如下：

```
void FindMinNode(BiTNode *b, char &min)
{ //调用前，初始化 min=b->data;
  if(b != NULL) {
    if(b->data < min)
      min = b->data;
    FindMinNode(b->lchild, min); //在左子树中找最小结点值
    FindMinNode(b->rchild, min); //在右子树中找最小结点值
  }
}
```

将求二叉树T的单分支DegreeOne()代码、求最小值FindMinNode代码添加到主函数main（）中，运行结果如下：

```
"G:\Dev-C++_Code\数据结构 严蔚敏 code\bt.exe"

建立二叉树，请输入结点值系列：
EBC##DA#G##F###

先序遍历序列：
E B C D A G F

中序遍历序列：
C B A G D F E

后序遍历序列：
C G A F D B E

二叉树的深度为 High=5

二叉树中结点的个数为 num=7

二叉树中单分支结点个数为 num=2

二叉树中最小值为 min=A

Process returned 0 (0x0)   execution time : 3.562 s
Press any key to continue.
```



END