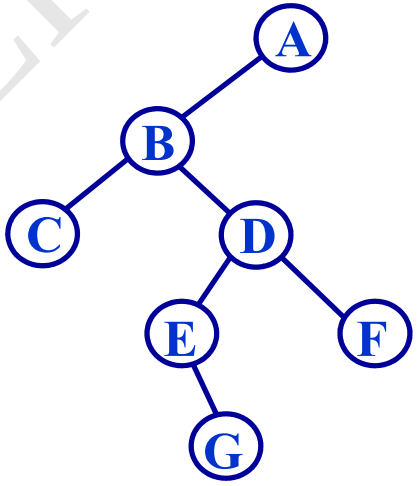
二叉树递归算法设计示例

QDU_JZR

【示例-1】课本中利用二叉树先序遍历序列建立二叉树的算法6.4。

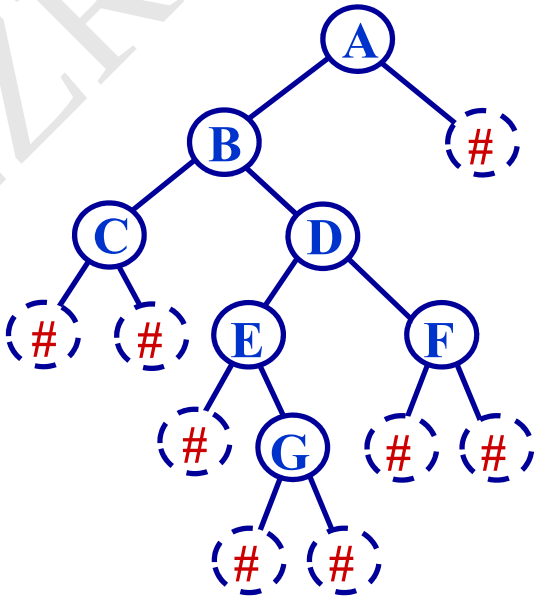
- 如欲建立如图所示的二叉树。
- 其先序序列为**A B C D E G F**。



QDU - LZR

【示例-1】课本中利用二叉树先序遍历序列建立二叉树的算法6.4。

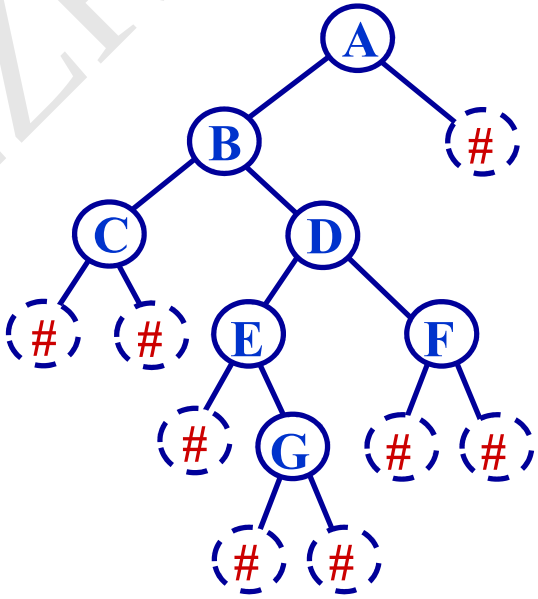
- 为了能够建立叶子结点，必须为二叉树添加一些“虚结点”。
- 对于如图所示的二叉树，约定以输入序列中不可能出现的值作为空结点的值以结束递归，例如用“#”、“$”等字符表示字符序列空结点。
- 按照先序遍历所得到的先序序列为A B C # # D E # G # # F # # #。



QDU LZR

【示例-1】课本中利用二叉树先序遍历序列建立二叉树的算法6.4。

- 算法的基本思想：
  - 每读入一个值，就为它建立一个结点，作为子树的根结点，其地址通过函数的引用型参数T直接链接到作为实际参数的指针中。
  - 然后，分别对根的左、右子树递归地建立子树，直到读入'#'建立空子树递归结束。



- 算法描述如下：

QU LZR

【示例-1】算法实现

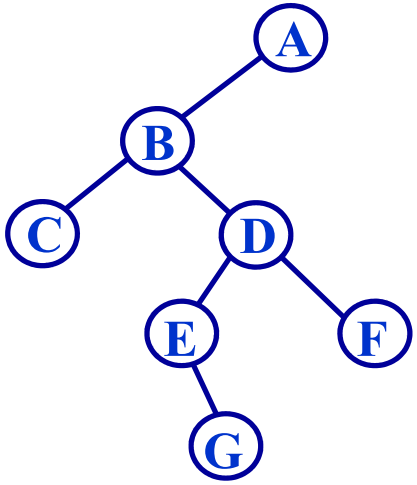
```
void CreateBiTree(BiTree &T)
{   //  算法6.4。按先序次序输入二叉树中结点的值
    //  构造二叉链表表示的二叉树T。
    TElemType   ch;
    scanf("%c", &ch);
    if(ch == '#')                  // 空树
        T = NULL;
    else {
        T = (BiTree)malloc(sizeof(BiTNode));
        if(!T)              // 检测是否申请结点成功
            exit(-1);
        T->data = ch;                 // 生成根结点
        CreateBiTree(T->lchild);   // 构造左子树
        CreateBiTree(T->rchild);   // 构造右子树
    }
}
```

ODU-LZR

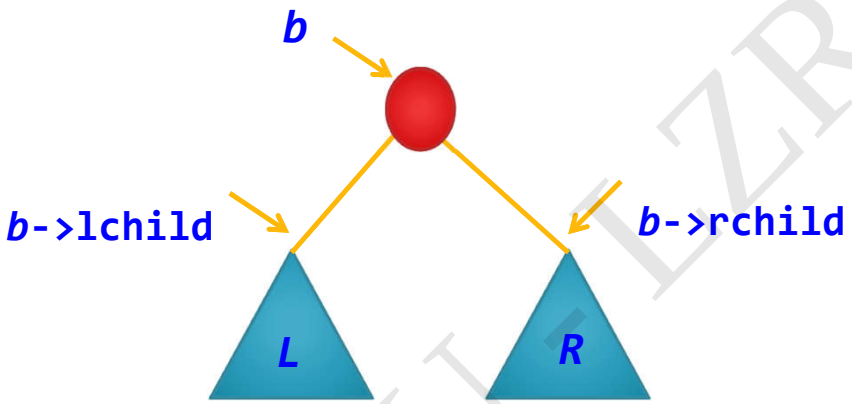
【示例-1】算法演示

```
#include <bits/stdc++.h>
using namespace std;
/* 其他代码 */
int
{
    InOrderTraversal(bt);
    printf("\n\n后序遍历序列: \n");
    PostOrderTraversal(bt);
    printf("\n\n");
}
```

```
"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:
ABC##DE#G##F###

先序遍历序列:
A B C D E G F

中序遍历序列:
C B E G D F A

后序遍历序列:
C G E F D B A
Process returned 0 (0x0)   execution time : 4.859 s
Press any key to continue.
```



【示例-2】设计求二叉树高度（深度）的算法。



求二叉树bt的高度的递归模型f(b)如下：

$$f(b)=0 \quad \text{若b=NULL}$$
$$f(b)=\max\{f(b\text{->lchild}),f(b\text{->rchild})\}+1 \quad \text{其他情况}$$

QDUTZR

求二叉树bt的高度的递归代码如下：

```
int bt_Height(BiTNode *T)
{
    if(T == NULL)
        return 0;
    else
        if(T->lchild == NULL && T->rchild == NULL)
            return 1;
        else
            return 1 + max(bt_Height(T->lchild),
                           bt_Height(T->rchild));
}
```

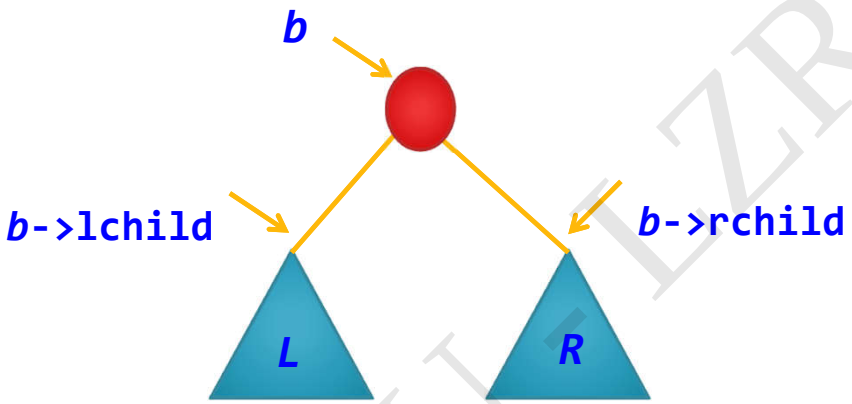
QDU - LZR

求二叉树bt的高度的递归代码如下：

```
int BTHeight(BiTNode *bt)
{  int lh,rh;
   if (bt==NULL)
      return 0;                          //空树的高度为0
   else
   {  lh=BTHeight(bt->lchild);           //求左子树的高度
      rh=BTHeight(bt->rchild);           //求右子树的高度
      return (lh>rh) ? (lh+1) : (rh+1);
   }
}
```

QDU - LZR

【示例-3】设计求二叉树结点个数算法。



求二叉树b的结点个数的递归模型f(b)如下：

```
f(b)=0                                  当bt=NULL
f(b)=f(b->lchild)+f(b->rchild)+1        其他情况
```

QDUTZR

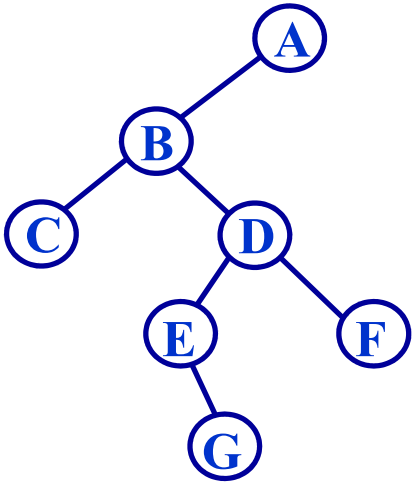
求二叉树T的结点个数的递归代码如下：

```
int NodeCount(BiTNode *T)                    //求二叉树T的结点个数
{
    int lnum, rnum;
    if(T == NULL)                            //为空树时返回0
        return 0;
    else {
        lnum = NodeCount(T->lchild); //求左子树结点个数
        rnum = NodeCount(T->rchild); //求右子树结点个数
        return (lnum + rnum + 1);    //返回和加上1
    }
}
```

QDU - LZR

将求二叉树T的高度代码、求结点个数的递归代码添加到主函数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：
ABC##DE#G##F###

先序遍历序列：
A B C D E G F

中序遍历序列：
C B E G D F A

后序遍历序列：
C G E F D B A

二叉树的深度为 H=5

二叉树中结点的个数为 Num=7

Process returned 0 (0x0)   execution time : 21.295 s
Press any key to continue.
```



END

QDU-LZR