

### //5.5.3 节 算法 DelSubtree

//在以 t 为根的树中删除根为 p 的子树

```
template<class T>
```

```
void Tree<T> :: DelSubtree (TreeNode<T> *t , TreeNode<T> *p )
```

```
{
    if ( t !=NULL && p != NULL ){ //指针 t 所指结点和指针 p 所指结点中有一个不存在，则返回
        TreeNode<T> *q=NULL,*result=NULL;
        result=FindFather(t , p);    //确定结点 p 的父结点是否存在
        if (result){
            if ((result->GetFirstChild()==p){    //若 p 的父结点存在，且 p 是其大儿子结点
                result->SetFirstChild (p->GetNextBrother());
                Del ( p );
                return;
            }
            else {
                q=result->GetFirstChild();
                while ((q->GetNextBrother())!= p)    q=q->GetNextBrother();
                q->SetNextBrother(p->GetNextBrother());
                Del ( p );
                return;
            }
        }
        else {    //result=NULL
            Del(p);
            root=NULL;
        }
    }
}
```