

#### //5.2.4节 算法LevelOrder

// 按层次次序遍历以指针t所指结点为根的树，利用一个辅助队列

template<class T>

void BinTree<T>:: LevelOrder (BinTreeNode<T> \*t ) const

```
{
    if(t==NULL) return;
    BinTreeNode<T> *p;
    AQueue<BinTreeNode<T> *> q ;           // AQueue为定义的队列
    if ( t!=NULL)
        q.QInsert ( t );                   // 根结点入队
    while (! q.IsEmpty ( ))
    {
        q.QDelete ( p );                   // 出队一个结点，设为当前结点
        cout<<p->GetData()<<endl;          // 访问当前结点
        if ( (p->GetLeft()) !=NULL ) q.QInsert (p->GetLeft() ); // 将当前结点的所有子结点入队
        if ( (p->GetRight())!=NULL ) q.QInsert (p->GetRight() );
    }
};
```