

## 05-H 层序遍历

#数据结构邓神

### 实现 借助队列实现层次遍历

```
// 层次遍历

template <typename T> template <typename VST> void BinNode<T>::travLevel(VST &
visit){
    queue<BinNodePosi(T)> Q;
    Q.push(this);
    while(!Q.empty()){
        BinNodePosi(T) x = Q.top();
        Q.pop();
        visit(x->data);
        if (HasLChild(*x)){
            Q.push(x->lchild);
        }
        if (HasRChild(*x)){
            Q.push(x->rchild);
        }
    }
}
```

#### 实现

```
❖ template <typename T> template <typename VST>
void BinNode<T>::travLevel( VST & visit ) { //二叉树层次遍历
    Queue<BinNodePosi(T)> Q; //引入辅助队列
    Q.enqueue( this ); //根节点入队
    while ( !Q.empty() ) { //在队列再次变空之前，反复迭代
        BinNodePosi(T) x = Q.dequeue(); //取出队首节点，并随即
        visit( x->data ); //访问之
        if ( HasLChild(*x) ) Q.enqueue( x->lChild ); //左孩子入队
        if ( HasRChild(*x) ) Q.enqueue( x->rChild ); //右孩子入队
    }
}
```

#### 实例

实例

