实验一 树和森林

计算机工程与科学学院 18120189 林艺珺

（一）验证性试验

1.树的双亲孩子表示法

在树的双亲孩子表示法中，设计并实现相应函数，完成相应二叉树的层次遍历。

在ChildParentTree.h的函数声明中添加：

void BinaryLevelOrder(void (\*Visit)(const ElemType &)) const;

在ChildParentTree.h的函数实现中添加：

template <class ElemType>

void ChildParentTree<ElemType>::BinaryLevelOrder(void (\*Visit)(const ElemType &)) const // 操作结果：层次遍历转换的二叉树

{

LinkQueue<int> q; // 队列

int cur = root; // 从根结点开始进行层次遍历

if (cur >= 0 || cur < num)

q.EnQueue(cur); // 如果根合法,则入队

while (!q.IsEmpty()) { // q非空,说明还有结点未访问

q.DelQueue(cur); // 出队

(\*Visit)(nodes[cur].data); // 访问当前结点

int fc = FirstChild(cur);

if (fc != -1) //队头元素第一个孩子非空

q.EnQueue(fc); //第一个孩子入队

int rs = RightSibling(cur);

if (rs != -1) //队头元素右兄弟非空

q.EnQueue(rs); //右兄弟孩子入队

}

}

在测试文件中，使用以下内容进行函数调用并输出结果：

char items[] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};

int parents[] = {-1, 0, 0, 0, 1, 1, 3, 3};

int r = 0, n = 8, cur;

ChildParentTree<char> t(items, parents, r, n);

cout << "层次遍历二叉树: ";

t.BinaryLevelOrder(Write);

cout << endl;

2.树的孩子兄弟表示法

在树的孩子兄弟链表示中，设计并实现相应函数，统计树中度数为1的结点数。

在ChildSiblingTree.h的函数声明中添加：

int DegreeOne();

在ChildSiblingTree.h的函数实现中添加：

template <class ElemType>

int ChildSiblingTree<ElemType>::DegreeOne()

{

int oneNum = 0; //度数为1结点计数器

ChildSiblingTreeNode<ElemType> \*p = root;

if (p->firstChild != NULL && p->firstChild->nextSibling == NULL)

{oneNum++;}

for (p = root->firstChild; p != NULL; p = p->nextSibling)

{ //遍历树

if (p->firstChild != NULL && p->firstChild->nextSibling == NULL) {oneNum++;}

}

return oneNum;

}

在测试文件中，使用以下内容进行函数调用并输出结果：

char items[] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'};

int parents[] = {-1, 0, 0, 0, 1, 1, 3, 3};

int r = 0, n = 8, cur;

ChildSiblingTree<char> t(items, parents, n);

cout << "度数为1的结点的数量: ";

cout << t.DegreeOne() << endl;