实验3实验日志

10月19日

- 开始对照书上的二叉链表ADT写二叉树的编写工作
- 遇到部分与书上代码块不同的区域，不过成功解决，完成二叉树的初步构建
- 代码正常编译，无报错（暂时没有进行样例的测试）

只实现了ADT的声明和定义，还没有实现ADT的功能

10月20日

- 尝试将ADT的声明和定义转为实现
- 根据小学期的由前中序求后序得到启发，写出由中后序得到前序，因此构建二叉树

```
node* node::create(int post[],int in[],int postL,int postR,int inL,int inR){
    if(postL > postR){
        return NULL;
    }
    node* root = new node;
    root->data = post[postR];
    int i;
    for(i = inL; i <= inR;i++ ){
        if(in[i] == post[postR]){
            break;
        }
    }
    int numLeft ;
    numLeft = i - inL;
    root->left = create(post,in,postL,postL+numLeft-1,inL,i-1);
    root->right = create(post,in,postL+numLeft,postR-1,i+1,inR);
    return root;
}
```

遇到问题如下：

- 开始使用vector存储动态数据，可是操作繁琐，最终失败
- 学习CSDN的方法，开辟全局数组（因为只有30位），取变量左右边界代替vector

解决问题： ADT实现，树的构建不报错成功编译

10月21日

- 完善二叉树的ADT，使其完成树的基本功能

```
node* create(int a[],int b[],int postL,int postR,int inL,int inR);
    void levelorder(node *root);
    void preOrder(node*tmp);
    void inOrder(node*tmp);
    void postOrder(node*tmp);
    int nodeDepth(node*tmp);
    int nodeNodes(node*tmp);
    int nodeHeight(node*tmp);
    int nodeLeafs(node*tmp);
    bool find(node*tmp, int e);
```

- 借此实现类内定义，类外实现的方法，简化了类的结构

难点

逐层输出二叉树的结构

难点在于，无法使用合适的递归函数完成，因为总会涉及到子节点的先后顺序输出问题。

尝试用多重循环判断，组合成while循环解题

最终发现，变成了另一种形式的递归（失败）

解决

询问了同学之后，在CSDN上找相关资料，发现用到了队列，借鉴CSDN的经验，写出如下代码

```
void node::levelorder(node *root){
    node *queue[31];
    int front =0, rear = 0;
    rear = (rear +1) % 31;
    queue[rear] = root;
    node *p;
    while(front != rear){
        front = (front + 1)%31;
        p = queue[front];
        printf("%d ", p->data);
        if(p->left != NULL){
            rear = (rear + 1) % 31;
            queue[rear] = p->left;
        }
        if(p->right != NULL){
            rear = (rear + 1) % 31;
            queue[rear] = p->right;
        }
    }
}
```

问题得到初步解决，OG测试通过

接下来解决部分警告，分别是

- 类外使用类内声明函数，没有加类的限定符
- 类的实例化，没有初始化类，（初定义）
- 数组传参，指针问题

了解到问题之后，很简单的问题，复习C++知识，得到解决

实验总结

- 通过这次实验，了解到，大部分知识相通，从前序中序到后序，后序中序到前序，要灵活应用
- ADT局限性要打破，树的ADT部分功能的实现，也用到了队列
- 通过这次实验，熟悉了队列，递归，了解了树的基本功能的实现原理