数据结构实验报告——二叉树的遍历

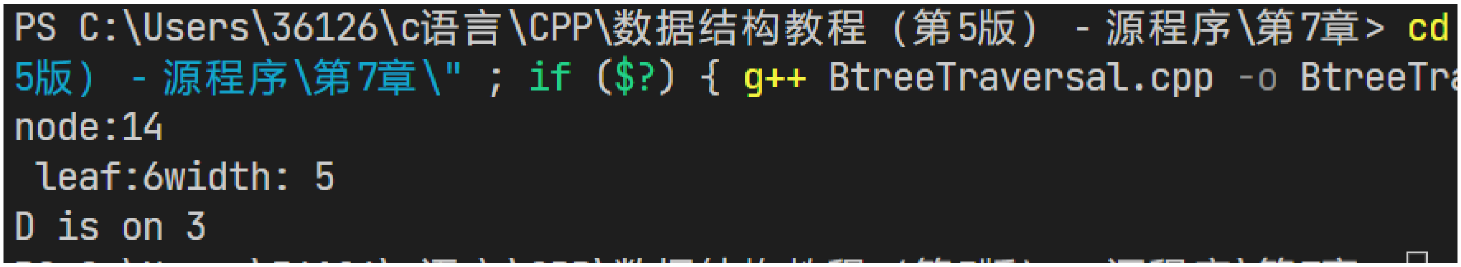
2022.10.26

2021100733 陈旭天

实验源码

```
#include"btree.cpp"
#include<queue>
using namespace std;

typedef struct {
    BTNode* ptr;
    int devi;
}TreeEle;

queue<TreeEle> q;

int nodeNum = 0;
int leaf = 0;

void PreTrave(BTNode* root){
    if(root)nodeNum++;// sum up node.
    if(root->lchild)PreTrave(root->lchild);
    if(root->rchild)PreTrave(root->rchild);
    if(root->lchild==nullptr&&root->rchild==nullptr)leaf++;//record leaf-node.
}

void InTrave(BTNode* root,int lr){
    if(root->lchild)InTrave(root->lchild,--lr);
    if(root){TreeEle the {root,lr};
    q.push(the);}
    if(root->rchild)InTrave(root->rchild,++lr);

}

void StageBtree(BTNode* root,ElemType x,int stage){
    if(root==nullptr)return ;
    if(root->data==x){printf("%c is on %d",x,stage+1);//output when we find it.}
    else if(root->lchild!=nullptr){
        StageBtree(root->lchild,x,++stage);
    }
    else if(root->rchild){
        StageBtree(root->rchild,x,++stage);
    }
}

int main(){
    char tree[] = "A(B(D,E(H(J,K(L,M(,N))))),C(F,G(,I)))";
    BTNode* test = nullptr;
    CreateBTree(test,tree);
    // DispBTree(test);
    PreTrave(test);
    printf("node:%d\n leaf:%d",nodeNum,leaf);
    InTrave(test,0);
    int left = 0;
    int right = 0;
    int q_size = q.size();
    int width = 0;
    for(int i = 0 ;i < q_size;++i){
        TreeEle a = q.front();
        left = min(a.devi,left);
        right = max(a.devi,right);
        q.push(q.front());
        q.pop();
    }
    width = right - left + 1;     //calculate the width of Btree.
    printf("width: %d\n",width);
    StageBtree(test,'D',0);
    return 0;
}
```

实验输出



```
PS C:\Users\36126\c语言\CPP\数据结构教程 （第5版） - 源程序\第7章> cd
5版） - 源程序\第7章\" ; if ($?) { g++ BtreeTraversal.cpp -o BtreeTra
node:14
 leaf:6width: 5
D is on 3
```