代码：（图片在下面）

#include <iostream>

#include <string>

using namespace std;

char ch;

int m = 0;

int n = 0;

int leaves = 0;

typedef struct TreeNode

{

char data;

struct TreeNode \*left;

struct TreeNode \*right;

}node, \*Node;

void DLR(Node &T)

{

if(T)

{

cout << T->data ;

DLR(T->left);

DLR(T->right);

}

}

void LDR(Node &T)

{

if(T)

{

LDR(T->left);

cout << T->data ;

LDR(T->right);

}

}

void LRD(Node &T)

{

if(T)

{

LRD(T->left);

LRD(T->right);

cout << T->data ;

}

}

void Create(Node &T) //先序建立二叉树

{

cin >> ch;

if(ch == '#')

{

T = NULL;

}

else

{

T = new node;

T->data = ch;

Create(T->left);

Create(T->right);

}

}

int Depth(Node &T)

{

if(T==NULL)

{

return 0;

}

else

{

m = Depth(T->left);

n = Depth(T->right);

if(m>n)

{

return (m+1);

}

else

{

return (n+1);

}

}

}

int NodeCount(Node &T)

{

if(T == NULL)

{

return 0;

}

else

{

return NodeCount(T->left) + NodeCount(T->right) + 1;

}

}

int Leaves(Node &T)

{

if(T)

{

if(T->left == NULL && T->right == NULL)

{

return leaves+1;

}

leaves = Leaves(T->left);

leaves = Leaves(T->right);

}

return leaves;

}

int Find(Node &T , char e)

{

if(T)

{

if(Find(T->left , e))

{

return 1;

}

if(T->data == e)

{

return 1;

}

if(Find(T->right , e))

{

return 1;

}

}

return 0;

}

void Check(Node &T , char e)

{

if(Find(T , e))

{

cout << "查找成功" << endl;

}

else

{

cout << "查找失败" << endl;

}

}

int main()

{

char find;

Node top = NULL;

cout << "创建二叉树" << endl;

Create(top);

cout << "先序遍历结果" << endl;

DLR(top);

cout << endl;

cout << "中遍历结果" << endl;

LDR(top);

cout << endl;

cout << "后序遍历结果" << endl;

LRD(top);

cout << endl;

cout << "结点总数：" << NodeCount(top) << endl;

cout << "叶子结点总数：" << Leaves(top) << endl;

cout << "深度：" << Depth(top) << endl;

cout << "请输入要查找的数据：";

cin >> find;

Check(top , find);

cout << "请输入要查找的数据：";

cin >> find;

Check(top , find);

}



