

《数据结构课程实践》

**实验报告**

**实验名称：二叉树的遍历**

**姓 名：杨天诏**

**学 号：20081831**

**专 业：通信工程**

**实验时间：11月21日**

**杭州电子科技大学**

**通信工程学院**

**（实验报告要求：填写三四五一共三部分内容，实验报告文件命名方式：*数据结构课程实践2022\_实验报告8-姓名-学号*，完成后上传电子版到校园网的网络教学平台）**

1. **实验目的**

**掌握二叉树遍历算法的设计和应用。**

1. **实验内容**
2. **（实验指导书143页实验7.2）实现二叉树遍历的先/中/后序遍历递归算法（跳过非遍历算法），以及层次遍历算法。**
3. **对于字符串Z(B(E,Y(G,X)),C(,D(I,J(,K))))所表示的二叉树，调用 CreateBTree()函数创建二叉树，编写算法找出值最大的叶子节点的字符（此例中值为X）。**
4. **算法设计**

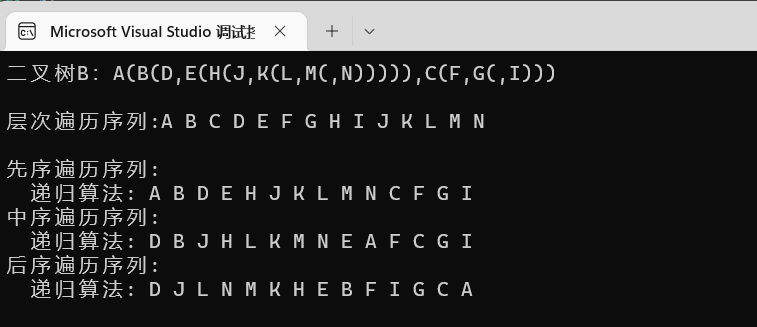
1. （可选）**对于三种递归算法，在适当的代码处设置断点，使用debugger调试器，打开watch窗口和call stack窗口，观察递归过程，记录截图，并加以必要的说明。**

**2. 给出附加题（第二题）的算法设计思路，如果必要，请画流程图或者示意图说明。**

函数输入为树与某一个叶子节点值，如果初始输入的树为NULL，则将初始输入的某一个叶子节点值返回；否则则进行判断：如果该节点的左右子树均不存在，则将该节点的值与初始值比对，将大的返回，如果存在左右子树，则将该函数进行遍历，以此来得到最后的最大叶子节点字符。

1. **运行结果与分析（给出运行结果截图，并加以简要说明）**

**题目一：**

****

**题目二：**

#include"x.h"

int MaxLeaf(BTNode\* b, char Max)

{

if (b == NULL)

{

return Max;

}

else

{

if (b->lchild == NULL && b->rchild == NULL)

{

if (b->data > Max)

Max = b->data;

return Max;

}

Max = MaxLeaf(b->lchild, Max);

Max = MaxLeaf(b->rchild, Max);

}

return Max;

}

int main()

{

BTNode\* b;

CreateBTree(b, "Z(B(E,Y(G,X)),C(,D(I,J(,K))))");

printf("二叉树b："); DispBTree(b); printf("\n");

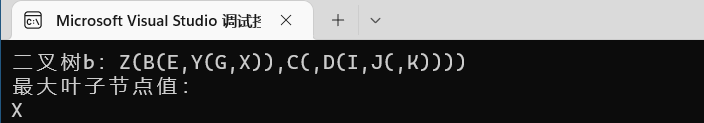
printf("最大叶子节点值：\n");

printf("%c", MaxLeaf(b, 'A'));

DestroyBTree(b);

return 1;

}

****

1. **实验小结（记录实验过程中遇到的主要问题和心得）**

**对二叉树遍历算法有了更深入的了解**