# 北京邮电大学软件学院

# 2015-2016学年第1学期实验报告

**课程名称： 算法与数据结构**

**实验名称： 树及其应用**

**实验完成人：**

**姓名：**\_徐朗朗\_\_\_**学号：**\_2016211965\_\_\_**成绩：**\_\_\_\_\_\_\_\_

**指导教师：**\_\_\_\_\_\_\_\_吴国仕\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**日 期： 2017 年 11 月 23 日**

1. **实验目的**

树是一种应用极为广泛的数据结构，也是这门课程的重点。它们的特点在于非线性。广义表本质上是树结构。本章实验继续突出了数据结构加操作的程序设计观点，但根据这两种结构的非线性特点，将操作进一步集中在遍历操作上，因为遍历操作是其他众多操作的基础。遍历逻辑的（或符号形式的）结构，访问动作可是任何操作。本次实验希望帮助学生熟悉各种存储结构的特征，以及如何应用树结构解决具体问题（即原理与应用的结合）。

1. **实验内容**

**必做内容**

* + 1. **二叉树的建立与遍历**

**[问题描述]**

　　建立一棵二叉树，并对其进行遍历（先序、中序、后序），打印输出遍历结果。

**[基本要求]**

　　从键盘接受输入（先序），以二叉链表作为存储结构，建立二叉树（以先序来建立），并采用递归算法对其进行遍历（先序、中序、后序），将遍历结果打印输出。

**[测试数据]**

ABCффDEфGффFффф（其中ф表示空格字符）则输出结果为

先序：ABCDEGF

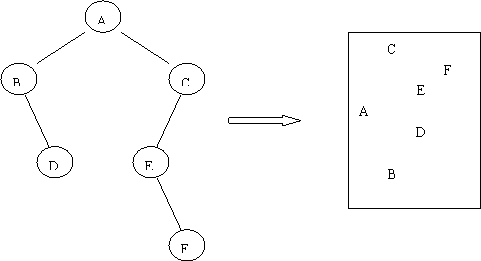
　　中序：CBEGDFA

　　后序：CGBFDBA

* + 1. **打印二叉树结构**

**[问题描述]**

　　按凹入表形式横向打印二叉树结构，即二叉树的根在屏幕的最左边，二叉树的左子树在屏幕的下边，二叉树的右子树在屏幕的上边。例如



**[测试数据]**

　　由学生依据软件工程的测试技术自己确定。注意测试边界数据，如空二叉树。

**[实现提示]**

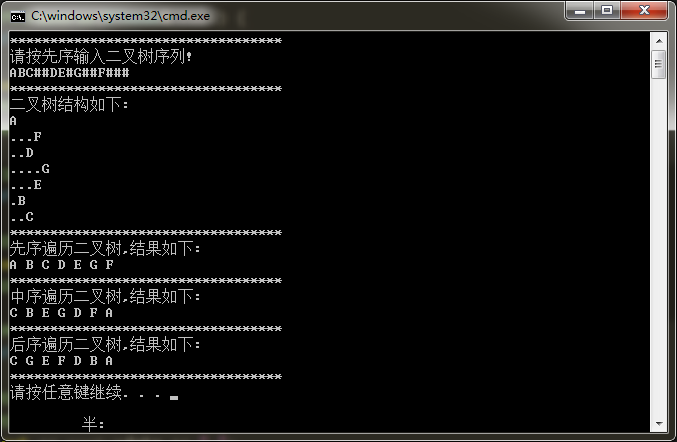
　　（1）利用RDL遍历方法；

（2）利用结点的深度控制横向位置。

1. **实验环境**

VS2017

1. **实验结果**



1. **附录**

关键代码展示: //二叉树节点定义

typedef struct node

{

char data; //数据

struct node\* lchild; //左孩子

struct node\* rchild; //右孩子

int depth; //深度

}\*BTnode;

//创建一棵二叉树

void createTree(BTnode& root, int nowdep) {

char c;

cin >> c;

if (c != '#')

{

root = new node();

root->data = c;

root->depth = nowdep;

}

else {

return;

}

//利用递归层数算深度

createTree(root->lchild, nowdep+1);

createTree(root->rchild, nowdep+1);

}