南京信息工程大学 数据结构 实验(实习)报告

实验(实习)名称 **二叉树的遍历** 实验(实习)日期 2021.11.19 得分 指导老师 马瑞

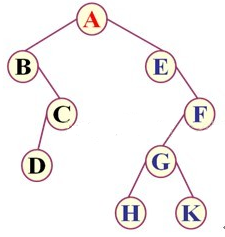
系 **计软院** 专业  **计算机科学与技术** 班级 20（1） 姓名 鲁哲豪 学号 202083290400

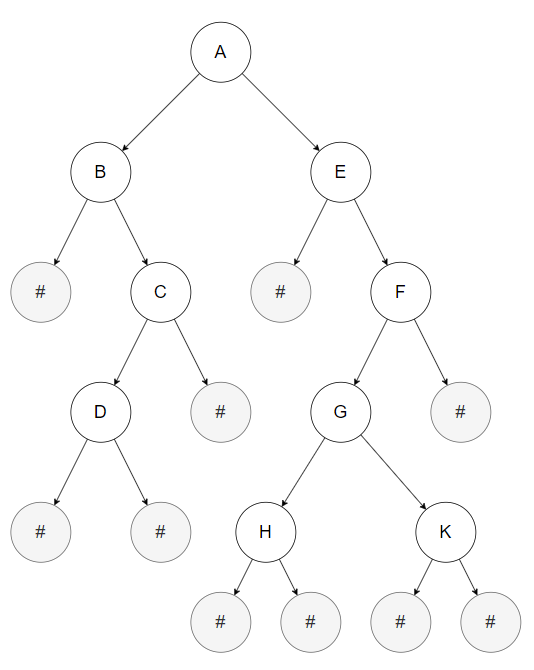
一、实验目的



二、 实验内容与步骤







如图



代码：

#include<stdio.h>

#include<stdlib.h>

typedef struct temp{

    char data;

    struct temp\* left,\* rignt;

} Node;

const char treedata[]=".ABE#C#F..D#..G#....##......HK..........................####....";

void pre(Node\* a,int pos){

    printf("%c",treedata[pos]);

    if(treedata[pos]!='#'){

        pre(a,pos\*2);

        pre(a,pos\*2+1);

    }

}

int main(){

    Node\* root=malloc(sizeof(Node));

    pre(root,1);

    return 0;

}

结果：





补全第2问的pre函数即可，无输出。修改部分的代码如下

void pre(Node\* a,int pos){

    a->data=treedata[pos];

    if(a->data!='#'){

        a->ls=malloc(sizeof(Node));

        a->rs=malloc(sizeof(Node));

        pre(a->ls,pos\*2);

        pre(a->rs,pos\*2+1);

    }else

        a->ls=a->rs=NULL;

}



递归：

新建函数in

void in(Node\* a){

    if(a->data=='#')

        return;

    in(a->ls);

    printf("%c",a->data);

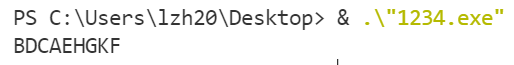
    in(a->rs);

}

在主函数调用

in(root);

结果如图：



非递归：

新建函数in2

void in2(Node\* a){

    Node\* stack[33]={};

    int top=1,back=0;

    while(top){

        if(!back && a->ls->data!='#'){

            stack[top++]=a;

            a=a->ls;

            continue;

        }

        putchar(a->data);

        back=0;

        if(a->rs->data!='#'){

            a=a->rs;

            continue;

        }

        a=stack[--top];

        back=1;

    }

}

在主函数调用

in2(root);

结果同上。



设节点个数为n，以上算法均为O(n)