

期末复习练习题1

开始时间2023/12/17 16:30:00

结束时间2024/01/13 18:30:00

答题时长39000分钟

答卷类型标准答案

总分120

填空题

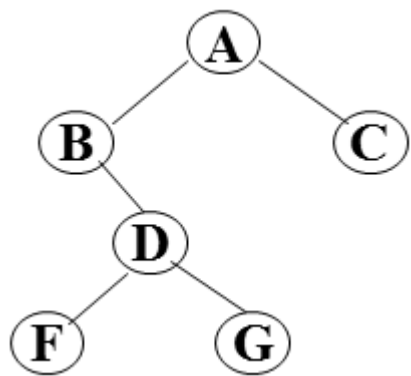
得分：暂无 总分：25

- 4-1 已知二叉树的先序遍历序列为
DAGICJBFHE
中序遍历序列为
GACIDFBHJE
则后序遍历序列为
GCIAFHBEJD (5分)
- 4-2 假设二叉树的存储结构为二叉链表，在具有n个结点的二叉链表中（n>0），左孩子指针域和右孩子指针域的个数为
2n|2*n|2 * n|n * 2|n*2 (2分)，空指针域的个数为 n+1|n + 1 (2分)。
- 4-3 如果二叉树的后序遍历结果是FDEBGCA，中序遍历结果是FDBEACG，那么该二叉树的前序遍历结果是：
ABDFECG|A B D F E C G (2分)。
- 4-4 一棵二叉树的先序序列：abdfcegh，中序序列：bfdagehc。后序遍历序列为（ ）。
fdbgheca|f d b g h e c a |f d b g h e c a (3分)
- 4-5 一棵二叉树的前序遍历序列是ABDFECGHK，中序遍历序列是DBEFAGHCK，则它的后序遍历序列是 DEFBHGKCA (3分)。(填写半角大写字母且不要添加空格，格式如ABCDEFG)。
- 4-6 将元素“50，30，15，35，70，65，95，60，25，40”依次插入开始为空的检索树，之后再逐一对它们进行查找，那么平均查找长度为 2.9|29/10 (4分)，删除元素50后，这棵检索树的根是 40 (4分)。

程序填空题

得分：暂无 总分：20

5-1 已知先序遍历序列和中序遍历序列建立二叉树。
例如



输入先序遍历序列：
ABDFGC，
再输入中序遍历序列：
BFDGAC，则
输出该二叉树的后序遍历序列：
FGDBCA。

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
typedef char ElementType;
typedef struct BiTNode{
    ElementType data;
    struct BiTNode *lchild;
    struct BiTNode *rchild;
}BiTNode,*BiTree;

BiTree CreatBinTree(char *pre,char *in,int n );
void postorder( BiTree T );

int main()
{
    BiTree T;
    char prelist[100];
    char inlist[100];
    int length;
    scanf("%s",prelist);
    scanf("%s",inlist);
    length=strlen(prelist);
    T=CreatBinTree(prelist,inlist, length);
    postorder( T );
    return 0;
}

void postorder( BiTree T )
{
    if(T)
    {
        postorder(T->lchild);
        postorder(T->rchild);
        printf("%c",T->data);
    }
}

BiTree CreatBinTree(char *pre,char *in,int n)
{
    BiTree T;
    int i;
    if(n<=0) return NULL;
    T=(BiTree)malloc(sizeof(BiTNode));
    T->data=pre[0];
    for(i=0;in[i]!=pre[0];i++);
    T->lchild= CreatBinTree(pre+1,in,i) (3分);
    T->rchild= CreatBinTree(pre+i+1,in+i+1,n-i-1) (3分);
    return T;
}

```

5-2 统计二叉树度为1的结点个数。

```

#include<iostream>
using namespace std;
typedef struct BiNode{
    char data;
    struct BiNode *lchild,*rchild;
}BiTNode,*BiTree;

```

```

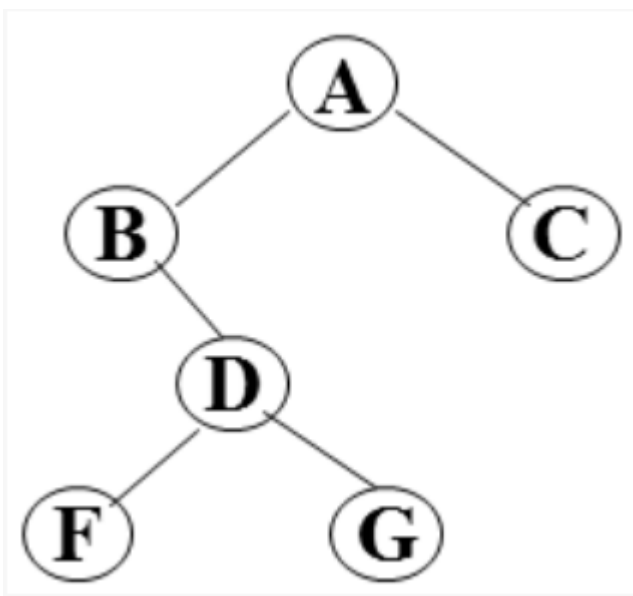
void CreateBiTree(BiTree &T){
    char ch;
    cin >> ch;
    if(ch=='#') T=NULL;
    else{
        T=new BiTNode;
        T->data=ch;
        CreateBiTree(T->lchild);
        CreateBiTree(T->rchild);
    }
}

int NodeCount ( BiTree T)
{
    if( T==NULL (2分)) return 0;
    if(T->lchild==NULL&&T->rchild!=NULL)
        return NodeCount(T->rchild)+1 (2分);
    if(T->lchild!=NULL&&T->rchild==NULL)
        return NodeCount(T->lchild)+1 (2分);
    return NodeCount(T->lchild)+NodeCount(T->rchild) (2分);
}

int main(){
    BiTree T;
    CreateBiTree(T);
    printf("%d", NodeCount(T));
    return 0;
}

```

输入样例1:



AB#DF##G##C##

输出样例1:

5-3 计算二叉树深度。

```
#include<iostream>
using namespace std;

typedef struct BiNode
{
    char data;
    struct BiNode *lchild,*rchild;
}BiTNode,*BiTree;

void CreateBiTree(BiTree &T)
{
    char ch;
    cin >> ch;
    if(ch=='#') T=NULL;
    else{
        T=new BiTNode;
        T->data=ch;
        CreateBiTree(T->lchild);
        CreateBiTree(T->rchild);
    }
}

int Depth(BiTree T)
{
    int m,n;
    if( T == NULL (2分)) return 0;
    else
    {
        m=Depth(T->lchild) (2分);
        n=Depth(T->rchild) (2分);
        if(m>n) return(m+1);
        else return (n+1);
    }
}

int main()
{
    BiTree tree;
    CreateBiTree(tree);
    cout<<Depth(tree);
    return 0;
}
```

编程题

得分：暂无 总分：75

7-1 玩转二叉树 (25分)

给定一棵二叉树的中序遍历和前序遍历，请你先将树做个镜面反转，再输出反转后的层序遍历的序列。所谓镜面反转，是指将所有非叶结点的左右孩子对换。这里假设键值都是互不相等的正整数。

输入格式：

输入第一行给出一个正整数 N (≤ 30)，是二叉树中结点的个数。第二行给出其中序遍历序列。第三行给出其前序遍历序列。数字间以空格分隔。

输出格式：

在一行中输出该树反转后的层序遍历的序列。数字间以1个空格分隔，行首尾不得有多余空格。

输入样例：

```
7
1 2 3 4 5 6 7
4 1 3 2 6 5 7
```

输出样例：

```
4 6 1 7 5 3 2
```

7-2 愿天下有情人都是失散多年的兄妹 (25分)

呵呵。大家都知道五服以内不得通婚，即两个人最近共同祖先如果在五代以内（即本人、父母、祖父母、曾祖父母、高祖父母）则不可通婚。本题就请你帮助一对有情人判断一下，他们究竟是否可以成婚？

输入格式：

输入第一行给出一个正整数 N ($2 \leq N \leq 10^4$)，随后 N 行，每行按以下格式给出一个人的信息：

```
本人ID 性别 父亲ID 母亲ID
```

其中 ID 是5位数字，每人不同；性别 M 代表男性、 F 代表女性。如果某人的父亲或母亲已经不可考，则相应的 ID 位置上标记为 -1 。

接下来给出一个正整数 K ，随后 K 行，每行给出一对有情人的 ID ，其间以空格分隔。

注意：题目保证两个人是同辈，每人只有一个性别，并且血缘关系网中没有乱伦或隔辈成婚的情况。

输出格式：

对每一对有情人，判断他们的关系是否可以通婚：如果两人是同性，输出 `Never Mind`；如果是异性并且关系出了五服，输出 `Yes`；如果异性关系未出五服，输出 `No`。

输入样例：

```
24
00001 M 01111 -1
00002 F 02222 03333
00003 M 02222 03333
00004 F 04444 03333
00005 M 04444 05555
00006 F 04444 05555
00007 F 06666 07777
00008 M 06666 07777
00009 M 00001 00002
00010 M 00003 00006
00011 F 00005 00007
00012 F 00008 08888
00013 F 00009 00011
00014 M 00010 09999
```

```
00015 M 00010 09999
00016 M 10000 00012
00017 F -1 00012
00018 F 11000 00013
00019 F 11100 00018
00020 F 00015 11110
00021 M 11100 00020
00022 M 00016 -1
00023 M 10012 00017
00024 M 00022 10013
9
00021 00024
00019 00024
00011 00012
00022 00018
00001 00004
00013 00016
00017 00015
00019 00021
00010 00011
```

输出样例：

```
Never Mind
Yes
Never Mind
No
Yes
No
Yes
No
No
```

鸣谢用户 徐校波 修正数据！

7-3 小字辈 (25分)

本题给定一个庞大家族的家谱，要请你给出最小一辈的名单。

输入格式：

输入在第一行给出家族人口总数 N （不超过 100 000 的正整数）—— 简单起见，我们把家族成员从 1 到 N 编号。随后第二行给出 N 个编号，其中第 i 个编号对应第 i 位成员的父/母。家谱中辈分最高的老祖宗对应的父/母编号为 -1 。一行中的数字间以空格分隔。

输出格式：

首先输出最小的辈分（老祖宗的辈分为 1，以下逐级递增）。然后在第二行按递增顺序输出辈分最小的成员的编号。编号间以一个空格分隔，行首尾不得有多余空格。

输入样例：

```
9
2 6 5 5 -1 5 6 4 7
```

输出样例：

4

1 9