第5章练习题

一. 单选题（共 35 分）

1. 前序遍历和中序遍历结果相同的二叉树是（ ）。（1 分）

A. 根结点无左孩子的二叉树　　B. 根结点无右孩子的二叉树

C. 所有结点只有左子树的二叉树　　D. 所有结点只有右子树的二叉树

2. 有一棵二叉树，其终端结点的个数为 7，度为 2 的结点个数应为（ ）（1 分）

A. 6　　B. 14

C. 8　　D. 9

3. 将一棵有 50 个结点的完全二叉树从 1 开始按层编号，则对编号为 25 的结点（）（1 分）

A. 无左、右孩子　　B. 有左孩子，无右孩子

C. 有右孩子，无左孩子　　D. 有左、右孩子

4. 二叉树是非线性数据结构，所以( )。（1 分）

A. 它不能用顺序存储结构存储；　　B. 它不能用链式存储结构存储；

C. 顺序存储结构和链式存储结构都能存储；　　D. 顺序存储结构和链式存储结构都不能使用

5. 一棵二叉树高度为 6，所有结点的度或为 0，或为 2，则这棵二叉树最少有( )结点。（1 分）

A. 12　　B. 11

C. 13　　D. 7

6. 如果结点 A 有 3 个同胞，B 是 A 的双亲，则结点 B 的度是（ ）（1 分）

1　　2

A . B .

C 3 . D 4 .

7. 树的后序遍历与对应二叉树的（ ）一致。（1 分）

A 前序遍历 . B 中序遍历 .

C 后序遍历 . D 层序遍历 .

8. 在按层次遍历二叉树的算法中，需要借助的辅助数据结构是（ ）（1 分）

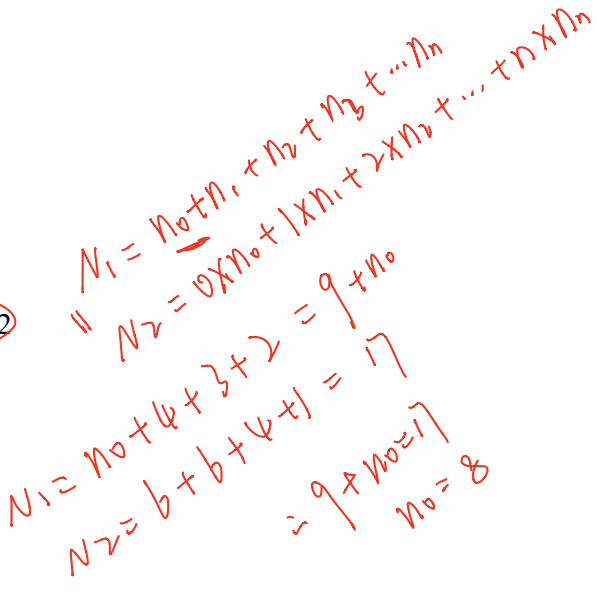
A 队列 . B 栈 .

C 线性表 . D 有序表 .

9. 已知一棵度为 3 的树有 2 个度为 3 的结点，3 个度为 2 的结点，4 个度为 1 的结点。则该树中有（ ）个叶结点。（1 分）

A 8 . B 10 .

C 12 . D 14 .



1 0. 设二叉树有 8 个结点，则其深度至少为（ ）（1 分）

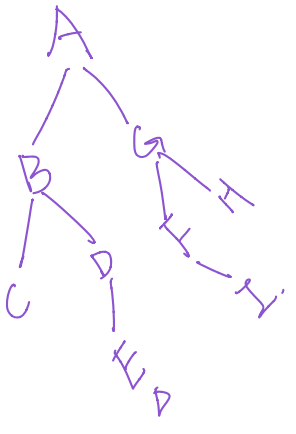
A 8 . B 6 .

C 4 . D 2 .

1 1. 已知二叉树的前序和中序分别为：ABCDEGFIH 和 CBEDAFIGH，则该二叉树后序遍历序列为（ ）（1 分）

A ABCDEFGHI . B IHGFEDCBA .

C CEDBIFHGA . D HFCDBGIEA .



1. 在线索二叉树中，一个结点是叶子结点的充要条件为（　　）（1分）

A. 左线索标志为0，右线索标志为1。

B. 左线索标志为1，右线索标志为0。

C. 左右线索标志均为0。

D. 左右线索标志均为1。

左0，右1.

3. 为5个使用频率不等的字段设计哈夫曼编码，不可能的设计方案是（ ）（1分）

A. 0000，0001，001，01，1

B. 000，001，01，10，11

C. 00，100，101，110，111

D. 000，001，010，011，1

4. 设哈夫曼编码的长度不超过4，若已经对两个字符编码为1和0，则最多还可以为（ ）个字符编码。（1分）

A. 5

B. 4

C. 3

D. 2

5. 设哈夫曼树中的叶子结点总数为m，若用二叉链表作为存储结构，则该哈夫曼树中总共有（ ）个空指针域。（1分）

A. 2m-1

B. 2m

C. 2m+1

D. 4m

6. 二叉树的深度为k，则二叉树最多有（ ）个结点。（1分）

A. 2k

B. 2（k-1） 注：（k-1)为上标

C. 2k-1 注：k为上标

D. 2k-1

7. 用顺序存储的方法，将完全二叉树中所有结点按层逐个从左到右的顺序存放在一维数组R[1……n]中，若结点R[i]有右孩子，则其右孩子是（ ）。（1分）

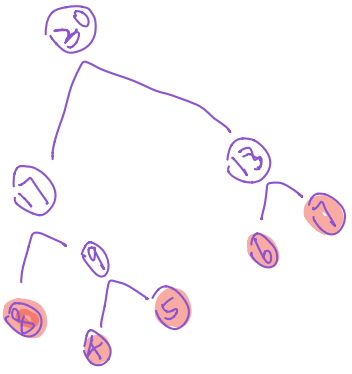
A. R[2i-1]

B. R[2i+1]

R[2i]

R[2/i]

C . D .



18. 若以 {4，5，6，7，8} 作为权值构造哈夫曼树，则该树的带权路径长度为（ ）。（1 分）

A 67 . B 68 .

C 69 . D 70 .

19. 将一棵有100个结点的完全二叉树进行层序编号，编号为49的结点的左孩子编号为（ ）。（1 分）

A 98 . B 99 .

C 50 . D 48 .

20. 树最适合用来表示（ ）。（1 分）

A 有序数据元素 . B 无序数据元素 .

C 元素之间具有分支层次关系的数据 . D 元素之间无联系的数据 .

21. 假定在一棵二叉树中，度为2的结点数为15，度为1的结点数为30，则叶子结点数为（ ）。（1 分）

A 15 . B 16 .

C 17 . D 47 .

22. 按照二叉树的定义，具有3个结点的二叉树有（ ）种。（1 分）

A 3 . B 4 .

C 5 . D 6 .

23. 把一棵树转换为二叉树后，这棵二叉树的形态是（）。（1 分）

A 唯一的 . B 有多种 .

有多种，但根结点没有左孩子　D．有多种，但根结点没有右孩子

C．

2　下列存储形式中，不是树的存储形式的是（）。（1 分）

4. A．双亲表示法　B．孩子链表表示法

C．孩子兄弟表示方法　D．顺序存储表示方法

2　讨论树，森林，二叉树的关系，目的是为了（）。（1 分）

5. A．借助二叉树上的运算方法实现对树的一些运算　B．将树，森林按照二叉树的存储方式进行存储并利用二叉树的算法解决树的有关问题

C．将树，森林转换成二叉树　D．体现一种技巧，没有什么实际意义

2　有关二叉树，下列说法正确的是（）。（1 分）

6. A．二叉树的度为2　B．一棵二叉树的度可以小于2度

C．二叉树中至少有一个结点的度为2　D．二叉树中任何一个结点的度都为2

2　用顺序存储方法将完全二叉树中的所有结点逐层存放在数组

7. R[1,2,...,n]中，结点R[i]若有左孩子，则左孩子的编号为结点（）。（1 分）

A．R[2i+1]　B．R[2i]

C．R[i/2]　D．R[2i-1]

2　一棵树的前序遍历序列为ABCDEFG，则它的中序遍历序列

8. 可能是（）。（1 分）

A．CABDEFG　B．ABCDEFG

C．DACEFBG　D．ABDCFEG

2　下列叙述正确的是（）。（1 分）

9. 二叉树是特殊的树　二叉树等价于度为2的树

A . B .

C 完全二叉树必为满二叉树 D 二叉树的左右子树有次序之分

3 一棵度为3的树中，度为3的结点数为2个，度为2的结点数为1
0. 个，度为1的结点数为2个，则度为0的结点数为（）个。（1 分）

A 4 B 5

C 6 D 7

3 在下述结论中，正确的是（）。 （1 分）
1. （1）只有一个结点的二叉树的度为0；
（2）二叉树的度为2；
（3）二叉树的左右孩子可以人以交换

(4)深度为K的完全二叉树的结点个数小于或等于深度相同的满二叉树；

A (1)(2)(3) B (2)(3)(4)

C (2)(4) D (1))(4)

3 一棵二叉树中，度为2的结点数为15个，度为1的结点数为30个，
2. 则度为0的结点数为（）个。（1 分）

A 15 B 16

C 17 D 47

3 一棵二叉树中，度为2的结点数为10个，度为1的结点数为5个，
3. 则叶子结点数为（）个。（1 分）

A 9 B 11

C 15 D 不确定

在一棵二叉树上第四层的结点数最多为（）个。（1 分）

A 2 B 4

3. C 6 D 8

3 在一棵二叉树上第i层的结点数最多为（）个。（1 分）

5. A 2i（i为上标） B 2i-1（i为上标）

C 2i+1（i为上标） D 2（i—1）（i-1为上标）

二. 多选题（共 6 分）

1. 中序遍历结果和后序遍历结果相同的二叉树有（ ）（2 分）

A 空二叉树 B 只有一个根结点的二叉树

C 左斜树 D 右斜树

2. 前序遍历结果和中序遍历结果相同的二叉树是（ ）。（2 分）

A 空二叉树 B 只有一个根结点的二叉树

C 左斜树 D 右斜树

3. 在下述结论中，正确的是（）。（2 分）

A 只有一个结点的二叉树的度为0 B 二叉树的度为2

C 二叉树的左右孩子可以人以交换 D 深度为K的完全二叉树的结点个数小于或等于深度相同的满二叉树

三. 判断题（共 24 分）

1. 哈夫曼树只有度为 0 和度为 2 的结点。（1 分）

2. 二叉链表表示的二叉树中，空指针域比非空指针域多 2 个。（1 分）

3. 编码集“00, 100, 101, 110, 111”是哈夫曼编码。（1 分）

4. 编码集“0，1，00，11”是前缀码（1 分）

5. 在二叉树的前序遍历中，任意一个结点均处在其子女的前面。（1 分）

6. 在二叉树的层次遍历中，任意一个结点均处在其子女的前面。（1 分）

7. 一棵满二叉树中共有 n 个结点，有 m 个叶子结点，树的深度为 h，则有 n = 2^h - 1。（1 分）
注：2^h 表示 2 的 h 次方。

8. 对于完全二叉树中的任一结点，若其右分支下的子孙的最大层次为 h，则其左分支下子孙的最大层次一定为 h + 1。（1 分）

9. 一棵有 n 个结点满二叉树，其中有 m 个叶子结点，则有 n = 2m - 1 成立。（1 分）

10. 设哈夫曼编码的长度不超过 4，若已对两个字符编码为 1 和 01，则最多还可以为 4 个字符编码。（1 分）

11. 在线索二叉树中，任一结点均有指向前趋和后继的线索。（1 分）

12. 二叉树是度为 2 的树。（1 分）

13. 由树转换成的二叉树，其根结点的右子树总是空的。（1 分）

14. 用一维数组存储二叉树，总是以前序遍历顺序存储结点。（1 分）

15. 已知一棵二叉树的前序和中序，则一定可以确定这棵二叉树。（1 分）

16. 对任一满二叉树，叶结点数为 m，则其结点总度数一定为 2(m - 1)。（1 分）

17. 树和二叉树之间具有一一对应的关系，因此，树和二叉树可以相互转换。（1 分）

18. 由 n 个权值构造的哈夫曼树一定有 n - 1 个分支结点。（1 分）

19. 对于一棵非空二叉树，它的根结点作为第一层，则它的第i层上最多能有2^i—1个结点。（1 分）
注：2^i 表示 2 的 i 次方。

20. 具有12个结点的完全二叉树有5个度为2的结点。（1 分）

21. 已知一棵二叉树的前序和后序遍历序列，可以唯一确定这棵二叉树。（1 分）

22. 存在这样的二叉树，对它采用任何次序的遍历，结果相同。（1 分）

23. 在哈夫曼编码中，当两个字符出现的频率相同时，其编码也相同。（1 分）

24. 完全二叉树的某结点若无左孩子，则它必须是叶子结点。（1 分）

四. 填空题（共 51 分）

1. 完全二叉树的高度为 5，则该二叉树最多有【1】个结点，至少有【2】个结点。（2 分）

2. 深度为 5 的二叉树最多有【1】个结点，最少有【2】个结点。（2 分）

3. 对于一棵具有 18 个结点的树，其所有结点的度之和为【1】。（1 分）

由权值为 {3, 4, 9, 2, 5}

4. 的叶子结点生成一棵哈夫曼树，其带权路径长度为【1】。（1分）

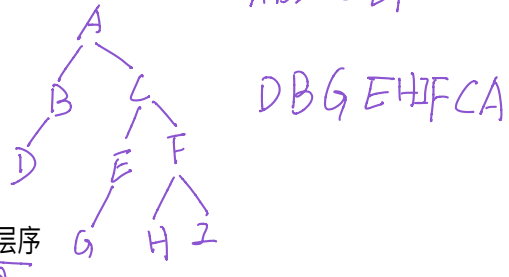
51

5. 假定一棵度为 3 的树中结点数为37，则其最小高度应为【1】，结点的最大度数为【2】。（2分）

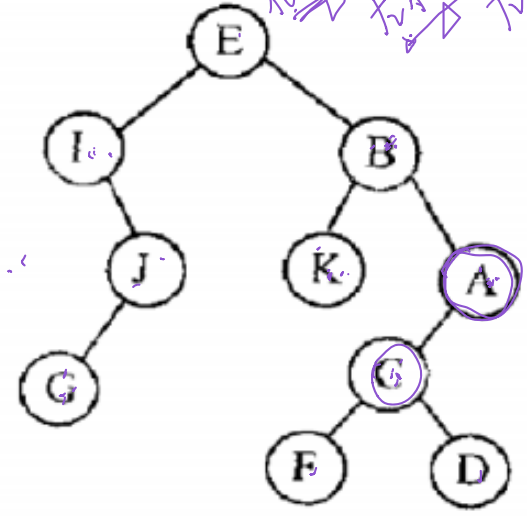
4 3

6. 现有某二叉树，按前序遍历的序列为ABDCEGFHI，按中序遍历的序列为BDAEGCHFI，写出此二叉树后序遍历序列。（1 分）

DBGEHIFCA



7. 写出如图所示的二叉树的前序遍历、中序遍历、后序和层序遍历序列。（4 分）

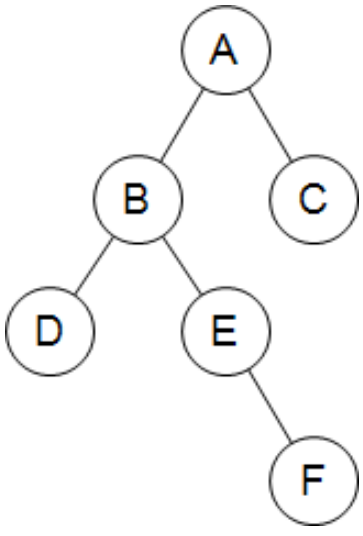


注意：全部连续填写大写字母

前序：【1】EIJGBKACFD
中序：【2】IGJEKBFCDA
后序：【3】GJIKFDCABE
层序：【4】EIBJKAGCFD

8. 已知如图所示二叉树的二叉链表用以下结构（BiTree）保存结点，root为二叉树根结点指针， 阅读函数 fun，写出执行fun(root)的返回值（ ）（1 分）

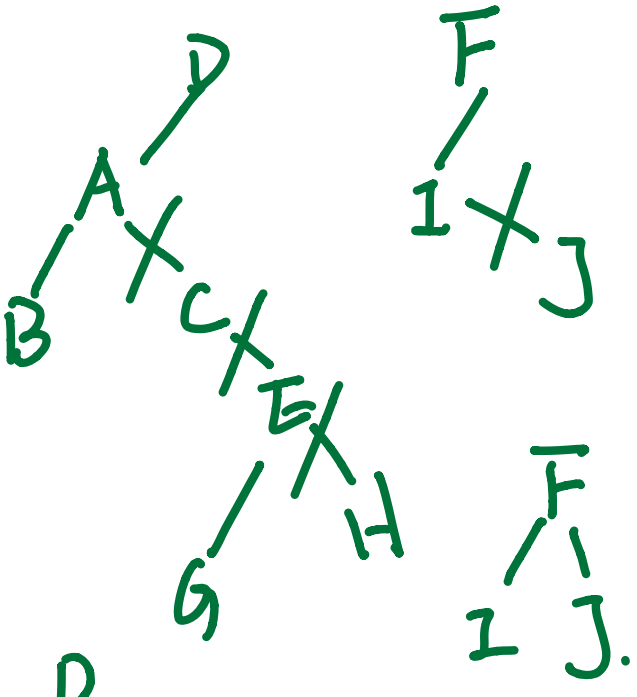
3

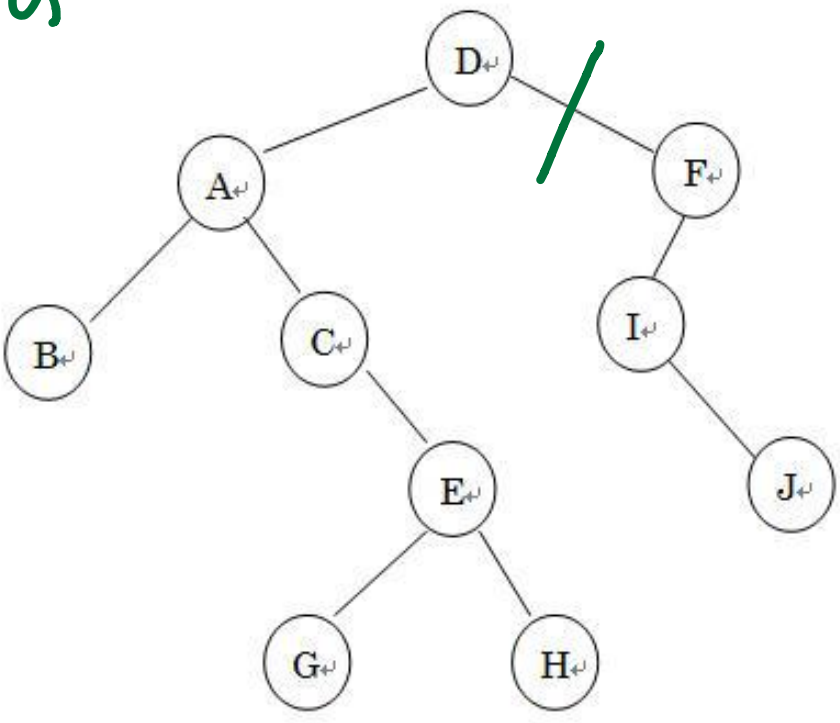



```
struct                                  BiTree
{
  char                  data;                   //数据域
  BinTree       *lchild,        *rchild;        //左右子树
};

int                     fun(BiTree                      *bt)
{
                        int             m,              n;
                if      (!bt)           return          0;
  if    (!bt->lchild    &&      !bt->rchild)    return  1;
  return                fun(bt->lchild)+fun(bt->rchild);
}
```

9. 已知二叉树如图所示，该二叉树转换为森林后有【1】棵树，森林中，最高的树的深度为【2】，最大度数的树的度为【3】。（3分）





1 已知一二叉链表的结点个数为
0. 13，则该二叉链表中有【1】个非空指针域，有【2】个空指针域。（2 分）

1 对于一棵具有 6
1. 个结点的树，其所有结点的度数之和为【1】，该树的度数最多为【2】。（2 分）

1 由权值为 {3, 8, 6, 2, 5}
2. 的叶子结点生成一棵哈夫曼树，其带权路径长度为【1】，树的深度为【2】，权值为 2 的叶结点的编码长度为【3】，权值为 6 的叶结点的编码长度为【4】。（4 分）

1 用顺序存储的方法将完全二叉树中的所有结点逐层存放到数
3. 组 A[1]~A[20] 中，对于层序编号为 9 的结点其根结点编号为【1】，若其有左子树，则左子树的根结点编号是【2】，若有右子树，则右子树的根结点编号是【3】。（3 分）

1 前序遍历结果为 ABC
4. 的二叉树共有【1】种可能，中序遍历为 ABC

的二叉树共有【2】种可能。（2 分）

1 完全二叉树第 5 层上有 6
5. 个叶结点，该完全二叉树至少有【1】个结点，最多有【2】个结点。（2 分）

1 5.设一棵完全二叉树的顺序存储结构中存储数据元素为ABC
6. DEF，则该二叉树的前序遍历序列为【1】，中序遍历序列为【2】，后序遍历序列为【3】。（3 分）

1 用5个权值{3, 2, 4, 5,
7. 1}构造的哈夫曼（Huffman）树的带权路径长度是（ ）。（1 分）

1 函数depth实现返回二叉树的高度，请在空格处将算法补充完
8. 整。（2 分）

```
int depth(Bitree *t)
{
  if(t==NULL)
    return 0;
  else
  {
    hl=depth(t->lchild);
    hr=【1】;
   if(【2】)
      return hl+1;
    else
      return hr+1;
   }
}
```

注：不要增加多余的分号

1 函数InOrder实现中序遍历二叉树，请在空格处将算法补充完
9. 整。（2 分）

```
void InOrder(Bitree *root)
{
  if(root==NULL) return 0;
  else
   {
     【1】;
```

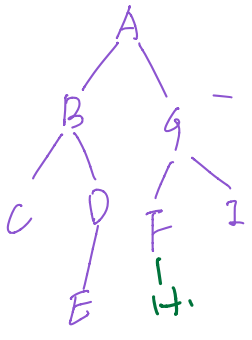
```
        cout<<root->data;
        【2】; InOrder(root->rchild)
    }
}
```

注：不要增加多余的分号

2 0. 函数PreOrderh实现非递归算法前序遍历二叉树，请在空格处将算法补充完整。（3 分）

```
void PreOrder(BiNode *root)
{
  top=-1;
  while(root!=NULL || top!=-1)
  {
    while(root!=NULL)
    {
      cout<<【1】; root->data
      s[++top]=root;
      root=root->lchild;
    }
    if(【2】) top!=-1
    {
      root=s[top--];
      root=【3】; root->rchild.
    }
  }
}
```

注：不要增加多余的分号 ABCDEGFIH



2 1. 已知二叉树的中序遍历序列为CBEDAFIGH，后序遍历序列为CEDBIFHGA，此二叉树的前序遍历序列为【1】。（1 分）

左根右 左右根

2 2. 在具有n个结点的二叉链表中，共有【1】个指针域，其中【2】个指针域用于指向其左右孩子，剩下的【3】个指针域则是空的。（3 分）

2n n-1 n+1

2 3. 树中某个结点的子树的个数称为该结点的【1】，子树的根结点称为该结点的【2】，该结点称为其子树根结点的【3】。（3 分）

度 孩子 双亲

24. 在具有n个叶子结点的huffman树中，叶子结点总数是n，分支结点总数是【1】。.（1 分）

n-1

练习题答案

一. 单选题 (共 35 分)

1.D 2.A 3.B 4.C 5.B 6.D 7.B 8.A 9.A 10.C 11.C 12.D 13.C 14.B 15.B
16.C 17.B 18.C 19.A 20.C 21.B 22.C 23.A 24.D 25.B 26.B 27.B 28.B 29.D 30.B
31.D 32.B 33.B 34.D 35.D

二. 多选题 (共 6 分)

1.ABC 2.AB 3.AD

三. 判断题 (共 24 分)

1.√ 2.√ 3.× 4.× 5.√ 6.√ 7.√ 8.× 9.√ 10.√ 11.× 12.×
13.√ 14.× 15.√ 16.√ 17.√ 18.√ 19.× 20.√ 21.× 22.√ 23.× 24.√

四. 填空题 (共 51 分)

1. 【1】 31

【2】 16

2. 【1】 31

【2】 5

3. 【1】 17

4. 【1】 51

5. 【1】 4

【2】 3

6. 【1】 DBGEHIFCA dbgehifca

7. 【1】 EIJGBKACFD eijgbkacfd

【2】 IGJEKBFCDA igjekbfcda

【3】 GJIKFDCABE gjikfdcabe

【4】 EIBJKAGCFD eibjkagcfd

8. 【1】 3

9. 【1】 2

【2】 3

【3】 4

10. 【1】 12

【2】 14

11. 【1】 5

【2】 5

12. 【1】 53

【2】 4

【3】 3

【4】 2

13. 【1】 4

【2】 18

【3】 19

14. 【1】 5

【2】 5

15. 【1】 21

【2】 51

16. 【1】 ABDECF

DBEAFC

【2】

【3】 DEBFCA

17. 【1】 33

18. 【1】 depth(t->rchild)

【2】 hl>hr

19. 【1】 InOrder(root->lchild)

【2】 InOrder(root->rchild)

20. 【1】 root->data

【2】 top!=-1

【3】 root->rchild

21. 【1】 ABCDEGFIH abcdegfih

22. 【1】 2n

【2】 n-1

【3】 n+1

23. 【1】 度 度数 结点度

【2】 孩子

【3】 双亲

24. 【1】 n-1