

得分

一、单项选择题 (X 题, 每题 X 分, 共 X 分)

评分标准: 每题回答正确得 1.5 分, 错误不得分!

- 1、深度为 k 的完全二叉树, 其叶子结点必在第 (C) 层上。
A. $k-1$ B. k C. $k-1$ 和 k D. 1 至 k
- 2、具有 60 个结点的二叉树, 其叶子结点有 12 个, 则度为 1 的结点数为 (D)。
A. 11 B. 13 C. 48 D. 37
- 3、Huffman 树的带权路径长度 WPL 等于 (C)。
A. 除根结点之外的所有结点权值之和 B. 所有结点权值之和
C. 各叶子结点的带权路径长度之和 D. 根结点的值
- 4、树形结构是数据元素之间存在一种 (D)。
A. 一对一关系 B. 多对多关系 C. 多对一关系 D. 一对多关系
- 5、在一棵度为 3 的树中, 度为 3 的结点数为 2 个, 度为 2 的结点数为 1 个, 度为 1 的结点数为 2 个, 则度为 0 的结点数为 (C) 个。
A. 4 B. 5 C. 6 D. 7
- 6、假设在一棵二叉树中, 双分支结点数为 15, 单分支结点数为 30 个, 则叶子结点数为 () (B) 个。
A. 15 B. 16 C. 17 D. 47
- 7、假定一棵三叉树的结点数为 50, 则它的最小高度为 (C)。
A. 3 B. 4 C. 5 D. 6
- 8、在一棵二叉树上第 4 层的结点数最多为 (D)。
A. 2 B. 4 C. 6 D. 8
- 9、用顺序存储的方法将完全二叉树中的所有结点逐层存放在数组中 $R[1..n]$, 结点 $R[i]$ 若有左孩子, 其左孩子的编号为结点 (B)。
A. $R[2i+1]$ B. $R[2i]$ C. $R[i/2]$ D. $R[2i-1]$
- 10、由权值分别为 3, 8, 6, 2, 5 的叶子结点生成一棵哈夫曼树, 它的带权路径长度为 (D)。
A. 24 B. 48 C. 72 D. 53
- 11、设 n, m 为一棵二叉树上的两个结点, 在中序遍历序列中 n 在 m 前的条件是 (B)。
A. n 在 m 右方 B. n 在 m 左方

C. n 是 m 的祖先

D. n 是 m 的子孙

12、如果 F 是由有序树 T 转换而来的二叉树，那么 T 中结点的前序就是 F 中结点的 (B)。

A. 中序

B. 前序

C. 后序

D. 层次序

13、欲实现任意二叉树的后序遍历的非递归算法而不必使用栈，最佳方案是二叉树采用 (A) 存储结构。

A. 三叉链表

B. 广义表

C. 二叉链表

D. 顺序

14、下面关于二叉树的叙述，正确的是 (D)。

A. 二叉树是特殊的树

B. 二叉树等价于度为 2 的树

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

15、任何一棵二叉树的叶子结点在先序、中序和后序遍历序列中的相对次序 (A)。

A. 不发生改变

B. 发生改变

C. 不能确定

D. 以上都不对

16、已知一棵完全二叉树的结点总数为 9 个，则最后一层的结点数为 (B)。

A. 1

B. 2

C. 3

D. 4

17、根据先序序列 ABDC 和中序序列 DBAC 确定对应的二叉树，该二叉树 (AD)。

A. 是完全二叉树

B. 不是完全二叉树

C. 是满二叉树

D. 不是满二叉树

18、树最适合用来表示 (C)。

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

19、二叉树的第 k 层的结点数最多为 (D)。

A. 2^{k-1}

B. $2k+1$

C. $2k-1$

D. 2^{k-1}

20、由权值分别为 11, 8, 6, 2, 5 的叶子结点生成一棵哈夫曼树，它的带权路径长度为 (B)。

A. 24

B. 71

C. 48

D. 53

21、在一棵度为 3 的树中，度为 3 的结点个数为 2，度为 2 的结点个数为 1，则度为 0 的结点个数为 (C)。

A. 4

B. 5

C. 6

D. 7

22、二叉树中第 $i(i \geq 1)$ 层上的结点数最多有 (C) 个。

A. $2i$

B. 2^i

C. 2^{i-1}

D. $2i-1$

23、设结点 A 有 3 个兄弟结点且结点 B 为结点 A 的双亲结点，则结点 B 的度为 (B)。

A. 3 B. 4 C. 5 D. 1

24、根据二叉树的定义可知二叉树共有 (B) 种不同的形态。

A. 4 B. 5 C. 6 D. 7

25、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 (B) 个空指针域。

A. $2m-1$ B. $2m$ C. $2m+1$ D. $4m$

26、设某棵二叉树的中序遍历序列为 ABCD，前序遍历序列为 CABD，则后序遍历该二叉树得到序列为 (A)。

A. BADC B. BCDA C. CDAB D. CBDA

27、设某棵二叉树中有 2000 个结点，则该二叉树的最小高度为 (C)。

A. 9 B. 10 C. 11 D. 12

28、设一棵二叉树的深度为 k ，则该二叉树中最多有 (D) 个结点。

A. $2k-1$ B. 2^k C. 2^{k-1} D. 2^k-1

29、设某二叉树中度数为 0 的结点数为 N_0 ，度数为 1 的结点数为 N_1 ，度数为 2 的结点数为 N_2 ，则下列等式成立的是 (C)。

A. $N_0=N_1+1$ B. $N_0=N_1+N_2$ C. $N_0=N_2+1$ D. $N_0=2N_1+1$

30、设一棵 m 叉树中度数为 0 的结点数为 N_0 ，度数为 1 的结点数为 N_1 ，……，度数为 m 的结点数为 N_m ，则 $N_0=$ (B)。

A. $N_1+N_2+\dots+N_m$ B. $1+N_2+2N_3+3N_4+\dots+(m-1)N_m$
C. $N_2+2N_3+3N_4+\dots+(m-1)N_m$ D. $2N_1+3N_2+\dots+(m+1)N_m$

31、设一组权值集合 $W=\{2, 3, 4, 5, 6\}$ ，则由该权值集合构造的哈夫曼树中带权路径长度之和为 (D)。

A. 20 B. 30 C. 40 D. 45

32、设二叉树的先序遍历序列和后序遍历序列正好相反，则该二叉树满足的条件是 (B)。

A. 空或只有一个结点 B. 高度等于其结点数
C. 任一结点无左孩子 D. 任一结点无右孩子

33、设某棵三叉树中有 40 个结点，则该三叉树的最小高度为 (B)。

A. 3 B. 4 C. 5 D. 6

34、深度为 k 的完全二叉树中最少有 (B) 个结点。

A. $2^{k-1}-1$ B. 2^{k-1} C. $2^{k-1}+1$ D. 2^k-1

- 35、设某哈夫曼树中有 199 个结点，则该哈夫曼树中有 (**B**) 个叶子结点。
- A. 99 B. 100 C. 101 D. 102
- 36、设按照从上到下、从左到右的顺序从 1 开始对完全二叉树进行顺序编号，则编号为 i 结点的左孩子结点的编号为 (**B**)。
- A. $2i+1$ B. $2i$ C. $i/2$ D. $2i-1$
- 37、设某棵二叉树的高度为 10，则该二叉树上叶子结点最多有 (**C**)。
- A. 20 B. 256 C. 512 D. 1024
- 38、设一棵完全二叉树中有 65 个结点，则该完全二叉树的深度为 (**B**)。
- A. 8 B. 7 C. 6 D. 5
- 39、设一棵三叉树中有 2 个度数为 1 的结点，2 个度数为 2 的结点，2 个度数为 3 的结点，则该三叉链权中有 (**C**) 个度数为 0 的结点。
- A. 5 B. 6 C. 7 D. 8
- 40、设 F 是由 T_1 、 T_2 和 T_3 三棵树组成的森林，与 F 对应的二叉树为 B ， T_1 、 T_2 和 T_3 的结点数分别为 N_1 、 N_2 和 N_3 ，则二叉树 B 的根结点的左子树的结点数为 (**A**)。
- A. N_1-1 B. N_2-1 C. N_2+N_3 D. N_1+N_3
- 41、设在一棵度数为 3 的树中，度数为 3 的结点数有 2 个，度数为 2 的结点数有 1 个，度数为 1 的结点数有 2 个，那么度数为 0 的结点数有 (**C**) 个。
- A. 4 B. 5 C. 6 D. 7
- 42、设某棵二叉树中只有度数为 0 和度数为 2 的结点且度数为 0 的结点数为 n ，则这棵二叉树中共有 (**C**) 个结点。
- A. $2n$ B. $n+1$ C. $2n-1$ D. $2n+1$
- 43、设一棵 m 叉树中有 N_1 个度数为 1 的结点， N_2 个度数为 2 的结点，……， N_m 个度数为 m 的结点，则该树中共有 (**D**) 个叶子结点。
- A. $\sum_{i=1}^m (i-1)N_i$ B. $\sum_{i=1}^m N_i$ C. $\sum_{i=2}^m N_i$ D. $1 + \sum_{i=2}^m (i-1)N_i$
- 44、设一组权值集合 $W=(15, 3, 14, 2, 6, 9, 16, 17)$ ，要求根据这些权值集合构造一棵哈夫曼树，则这棵哈夫曼树的带权路径长度为 (**D**)。
- A. 129 B. 219 C. 189 D. 229

得分

二、填空题 (**X** 题，每题 **X** 分，共 **X** 分)

评分标准：每空回答正确得 1 分，错误不得分，不完全正确酌情给分！

- 1、已知二叉树的中序遍历序列为 BCA，后序遍历序列为 CBA，则该二叉树的先序遍历序列为 【1】ABC，层序遍历序列为 【2】ABC。
- 2、在树型结构中，树根结点没有 【3】前趋 结点，其余每个结点的有且只有 【4】一 个前趋驱结点；叶子结点没有 【5】后继 结点；其余每个结点的后续结点可以有 【6】多 个。
- 3、设 F 是一个森林，B 是由 F 转换得到的二叉树，F 中有 n 个非终端结点，则 B 中右指针域为空的结点有 【7】n+1 个。
- 4、对于一个有 n 个结点的二叉树，当它为一棵 【8】完全 二叉树时具有最小高度，即为 【9】 $\lfloor \log_2 n \rfloor + 1$ ，当它为一棵单支树具有 【10】最大 高度，即为 【11】n。
- 5、由带权为 3, 9, 6, 2, 5 的 5 个叶子结点构成一棵哈夫曼树，则带权路径长度为 【12】55。
- 6、在一棵二叉排序树上按 【13】中序 遍历得到的结点序列是一个有序序列。
- 7、对于一棵具有 n 个结点的二叉树，当进行链接存储时，其二叉链表中的指针域的总数为 【14】2n 个，其中 【15】n-1 个用于链接孩子结点，【16】n+1 个空闲着。
- 8、在一棵二叉树中，度为 0 的结点个数为 n_0 ，度为 2 的结点个数为 n_2 ，则 $n_0 =$ 【17】 $n_2 + 1$ 。
- 9、一棵深度为 k 的满二叉树的结点总数为 【18】 $2^k - 1$ ，一棵深度为 k 的完全二叉树的结点总数的最小值为 【19】 2^{k-1} ，最大值为 【20】 $2^k - 1$ 。
- 10、由三个结点构成的二叉树，共有 【21】5 种不同的形态。
- 11、设高度为 h 的二叉树中只有度为 0 和度为 2 的结点，则此类二叉树中所包含的结点数至少为 【22】 $2h - 1$ 。
- 12、一棵含有 n 个结点的 k 叉树，【23】单支树 形态达到最大深度，【24】完全二叉树 形态达到最小深度。
- 13、对于一棵具有 n 个结点的二叉树，若一个结点的编号为 i ($1 \leq i \leq n$)，则它的左孩子结点的编号为 【25】 $2i$ ，右孩子结点的编号为 【26】 $2i + 1$ ，双亲结点的编号为 【27】 $i/2$ 。
- 14、对于一棵具有 n 个结点的二叉树，采用二叉链表存储时，链表中指针域的总数为 【28】 $2n$ 个，其中 【29】n-1 个用于链接孩子结点，【30】n+1 个空闲着。
- 15、哈夫曼树是指 【31】带权路径长度最小 的二叉树。
- 16、空树是指 【32】结点数为 0，最小的树是指 【33】只有一个根结点的树。
- 17、二叉树的链式存储结构有 【34】二叉链表 和 【35】三叉链表 两种。
- 18、三叉链表比二叉链表多一个指向 【36】双亲结点 的指针域。
- 19、数据结构中树的存储结构有 【37】双亲表示法、【38】孩子[链表]表示法 和

【39】长子兄弟表示法（孩子兄弟表示法）。

- 20、假定一棵树的广义表表示为 $A(B(E), C(F(H, I, J), G), D)$ ，则该树的度为【40】3，树的深度为【41】4，终端结点的个数为【42】6，单分支结点的个数为【43】1，双分支结点的个数为【44】1，三分支结点的个数为【45】2，C 结点的双亲结点为【46】A，其孩子结点为【47】F和【48】G结点。
- 21、一棵结点数为 n 的二叉树，其所有结点的度的总和是【49】 $n-1$ 。
- 22、对一棵二叉搜索树进行中序遍历时，得到的结点序列是一个【50】有序序列。对一棵由算术表达式组成的二叉语法树进行后序遍历得到的结点序列是该算术表达式的【51】后缀表达式（或逆波兰式）。
- 23、对于一棵具有 n 个结点的二叉树，用二叉链表存储时，其指针总数为【52】 $2n$ 个，其中【53】 $n-1$ 个用于指向孩子，【54】 $n+1$ 个指针是空闲的。
- 24、若对一棵完全二叉树从 0 开始进行结点的编号，并按此编号把它顺序存储到一维数组 A 中，即编号为 0 的结点存储到 $A[0]$ 中。其余类推，则 $A[i]$ 元素的左孩子元素为【55】 $2i+1$ ，右孩子元素为【56】 $2i+2$ ，双亲元素为【57】 $(i-1)/2$ 。
- 25、假定一棵树的广义表表示为 $A(C, D(E, F, G), H(I, J))$ ，则树中所含的结点数为【58】9个，树的深度为【59】3，树的度为【60】3。
- 26、若用链表存储一棵二叉树时，每个结点除数据域外，还有指向左孩子和右孩子的两个指针。在这种存储结构中， n 个结点的二叉树共有【61】 $2n$ 个指针域，其中有【62】 $n-1$ 个指针域是存放了地址，有【63】 $n+1$ 个指针是空指针。
- 27、一棵高度为 5 的二叉树中最少含有【64】5个结点，最多含有【65】31个结点；
- 28、假定一棵树的广义表表示为 $A(B(C, D(E, F, G), H(I, J)))$ ，则树中所含的结点数为【66】10个，树的深度为【67】4，树的度为【68】3，结点 H 的双亲结点为【69】B，结点 H 的孩子结点为【70】I 和 J。
- 29、已知一棵完全二叉树中共有 768 结点，则该树中共有【71】384个叶子结点。
- 30、设一棵二叉树中有 n 个结点，则当用二叉链表作为其存储结构时，该二叉链表中共有【72】 $2n$ 个指针域，【73】 $n+1$ 个空指针域。
- 31、设一棵二叉树的前序遍历序列和中序遍历序列均为 ABC ，则该二叉树的后序遍历序列为【74】CBA。
- 32、设一棵完全二叉树中有 21 个结点，如果按照从上到下、从左到右的顺序从 1 开始顺序编号，则编号为 8 的双亲结点的编号是【75】4，编号为 8 的左孩子结点的编号是【76】16。
- 33、中序遍历二叉排序树所得到的序列是【77】有序序列（填有序或无序）。
- 34、设某棵二叉树中度数为 0 的结点数为 N_0 ，度数为 1 的结点数为 N_1 ，则该二叉树中度数为

- 2 的结点数为 【78】 N_0-1 ；若采用二叉链表作为该二叉树的存储结构，则该二叉树中共有 【79】 $2N_0+N_1$ 个空指针域。
- 35、设一棵完全二叉树中有 500 个结点，则该二叉树的深度为 【80】 9；若用二叉链表作为该完全二叉树的存储结构，则共有 【81】 501 个空指针域。
- 36、设哈夫曼树中共有 n 个结点，则该哈夫曼树中有 【82】 0 个度数为 1 的结点。
- 37、设有 n 个结点的完全二叉树，如果按照从自上到下、从左到右从 1 开始顺序编号，则第 i 个结点的双亲结点编号为 【83】 $i/2$ ，右孩子结点的编号为 【84】 $2i+1$ 。
- 38、深度为 k 的完全二叉树中最少有 【85】 2^{k-1} 个结点。
- 39、设哈夫曼树中共有 99 个结点，则该树中有 【86】 50 个叶子结点；若采用二叉链表作为存储结构，则该树中有 【87】 100 个空指针域。
- 40、设前序遍历某二叉树的序列为 ABCD，中序遍历该二叉树的序列为 BADC，则后序遍历该二叉树的序列为 【88】 BDCA。
- 41、设一棵完全二叉树的顺序存储结构中存储数据元素为 ABCDEF，则该二叉树的前序遍历序列为 【89】 ABDECF，中序遍历序列为 【90】 DBEAF C，后序遍历序列为 【91】 DEBFCA。
- 42、设一棵完全二叉树有 128 个结点，则该完全二叉树的深度为 【92】 8，有 【93】 64 个叶子结点。
- 43、设二叉树中度数为 0 的结点数为 50，度数为 1 的结点数为 30，则该二叉树中总共有 【94】 129 个结点数。
- 44、设二叉树中结点的两个指针域分别为 lchild 和 rchild，则判断指针变量 p 所指向的结点为叶子结点的条件是 【95】 $p->lchild==0 \ \&\& \ p->rchild==0$ 。
- 45、设一棵三叉树中有 50 个度数为 0 的结点，21 个度数为 2 的结点，则该二叉树中度数为 3 的结点数有 【96】 14 个。
- 46、高度为 h 的完全二叉树中最少有 【97】 2^{h-1} 个结点，最多有 【98】 2^h-1 个结点。
- 47、设某棵二叉树的中序遍历序列为 ABCD，后序遍历序列为 BADC，则其前序遍历序列为 【99】 CABD。
- 48、完全二叉树中第 5 层上最少有 【100】 1 个结点，最多有 【101】 16 个结点。
- 49、设某棵完全二叉树中有 100 个结点，则该二叉树中有 【102】 50 个叶子结点。
- 50、设一棵二叉树的中序遍历序列为 BDCA，后序遍历序列为 DBAC，则这棵二叉树的前序序列为 【103】 CBDA。
- 51、设用于通信的电文仅由 8 个字母组成，字母在电文中出现的频率分别为 7、19、2、6、32、3、21、10，根据这些频率作为权值构造哈夫曼树，则这棵哈夫曼树的高度为 【104】 6。

52、设一棵 m 叉树的结点数为 n ，用多重链表表示其存储结构，则该树中有 【105】 $n(m-1)+1$ 个空指针域。

53、下面程序段的功能是建立二叉树的算法，请在下划线处填上正确的内容。

```
typedef struct node {
    int data;
    struct node *lchild;
    【106】 struct node *rchild;
} bitree;

void createbitree(bitree *&bt) {
    scanf("%c",&ch);
    if(ch=='#') 【107】 bt=0;
    else {
        bt=(bitree*)malloc(sizeof(bitree));
        bt->data=ch;
        【108】 createbitree(bt->lchild);
        createbitree(bt->rchild);
    }
}
```

得分

三、判断题（**X** 题，每题 **X** 分，共 **X** 分。正确填 ‘T’，错误 “F”。）

评分标准：每题回答正确得 2 分，错误不得分！

- 1、二叉树中每个结点的度不能超过 2，所以二叉树是一种特殊的树。 (F)
- 2、二叉树的前序遍历中，任意结点均处在其子女结点之前。 (T)
- 3、哈夫曼树的总结点个数（多于 1 时）不能为偶数。 (T)
- 4、由二叉树的先序序列和后序序列可以唯一确定一颗二叉树。 (F)
- 5、树的后序遍历与其对应的二叉树的后序遍历序列相同。 (F)
- 6、根据任意一种遍历序列即可唯一确定对应的二叉树。 (F)
- 7、满二叉树也是完全二叉树。 (T)
- 8、哈夫曼树一定是完全二叉树。 (F)
- 9、树的子树是无序的。 (F)
- 10、满二叉树一定是完全二叉树，完全二叉树不一定是满二叉树。 (T)

- 11、设一棵二叉树的先序序列和后序序列，则能够唯一确定出该二叉树的形状。 (F)
- 12、设一棵树 T 可以转化成二叉树 BT，则二叉树 BT 中一定没有右子树。 (T)
- 13、中序遍历二叉排序树可以得到一个有序的序列。 (T)
- 14、完全二叉树中的叶子结点只可能在最后两层中出现。 (T)
- 15、哈夫曼树中没有度数为 1 的结点。 (T)
- 16、先序遍历一棵二叉排序树得到的结点序列不一定是有序的序列。 (T)
- 17、由树转化成二叉树，该二叉树的右子树不一定为空。 (F)
- 18、若一个叶子结点是某二叉树的中序遍历序列的最后一个结点，则它必是该二叉树的先序遍历序列中的最后一个结点。 (T)
- 19、中序遍历一棵二叉排序树可以得到一个有序的序列。 (T)

得分

四、阅读程序题 (X 题，每题 X 分，共 X 分)

评分标准：每空回答正确得 1 分，错误不得分，不完全正确则酌情给分！

- 1、阅读程序回答问题。(评分标准：第 1 点 4 分，第 2 点 1 分！)

```
void ABC(BTNode * BT) {
    if(BT) {
        ABC(BT->left);
        ABC(BT->right);
        cout<<BT->data<<' ';
    }
}
```

(1) 该算法的功能是什么？

答：(1) 递归地后序遍历链式存储的二叉树。

- 2、阅读以下二叉树操作算法，指出该算法的功能。(评分标准：第 1 点 4 分，第 2 点 1 分！)

```
Template <calss type> void BinTree <Type>::unknown (BinTreeNode<Type>*t) {
    BinTreeNode< Type> *p=t, *temp;
    if (p!=NULL) {
        temp = p->leftchild;
        p->leftchild = p->rightchild;
        p->rightchild = temp;

        unknown(p->leftchild);
        undnown(p->rightchild);
    }
}
```

}

答：该算法的功能是：交换二叉树的左右子树的递归算法；

3、已知二叉树的存储结构为二叉链表，对应的二叉树如图，阅读程序解答问题：（1）画出执行上述算法后所建立的结构；（2）说明该算法的功能。（评分标准：第1点4分，第2点1分！）

```
typedef struct node {  
    DateType data;  
    Struct node * next;  
} ListNode;
```

```
typedef ListNode * LinkList;
```

```
LinkList Leafhead=NULL;
```

```
Void Inorder(BinTree T) {
```

```
    LinkList s;
```

```
    if(T){
```

```
        Inorder(T->lchild);
```

```
        if( (!T->lchild) && (!T->rchild) ) {
```

```
            s=(ListNode*)malloc(sizeof(ListNode));
```

```
            s->data = T->data;
```

```
            s->next = Leafhead;
```

```
            Leafhead = s;
```

```
        }
```

```
        Inorder(T->rchild);
```

```
    }
```

```
}
```

答：（1）



（2）中序遍历二叉树，按遍历序列中叶子结点数据域的值构建一个以 Leafhead 为头指针的逆序单链表（或按二叉树中叶子结点数据自右至左链接成一个链表）；

得分

五、应用题（X 题，每题 X 分，共 X 分）

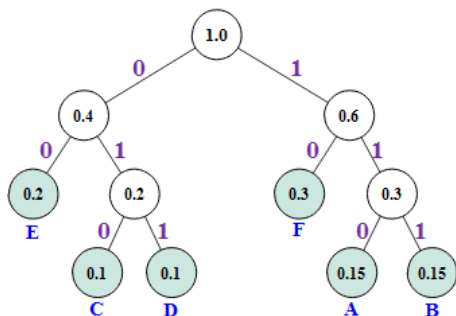
评分标准：每题回答完全正确得5分，其余按得分点给分！

1、已知数据六个字母及在通信中出现频率如下表，请把这些字母和频率作为叶子结点及权值，完成如下工作（要求有过程）。约定（1）权值小的二叉树作为新构造的二叉树的左子树；（2）在权值相等时，深度小的二叉树作为新构造的二叉树的左子树。（评分标准：共7分。

第1点4分, 第2点1分!)

A	B	C	D	E	F
0.15	0.15	0.1	0.1	0.2	0.3

- (1) 画出对应的 Huffman 树。
- (2) 计算带权路径长度 WPL。
- (3) 求 A、B、C、D、E、F 的 Huffman 编码。



答: (1)

(3 分)

(2) $WPL = 0.1 \times 3 + 0.1 \times 3 + 0.2 \times 2 + 0.15 \times 3 + 0.15 \times 3 + 0.3 \times 2 = 2.5$

(1 分)

(3) A: 110 B: 111 C: 010 D: 011 E: 00 F: 10

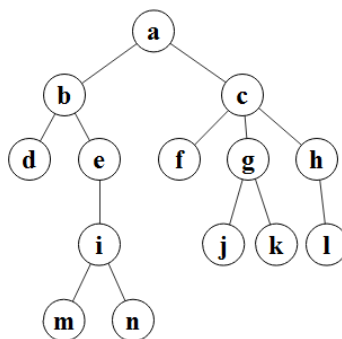
(3 分)

2、已知一棵树边的集合为{<i, m>, <i, n>, <e, i>, <b, e>, <b, d>, <a, b>, <g, j>, <g, k>, <c, g>, <c, f>, <h, l>, <c, h>, <a, c>}, 请画出这棵树, 并回答下列问题: (评分标准: 共5分。第1点4分, 第2点1分!)

- (1) 哪个是根结点?
- (2) 哪些是叶子结点?
- (3) 哪个是结点 g 的双亲?
- (4) 哪些是结点 g 的祖先?
- (5) 哪些是结点 g 的孩子?
- (6) 哪些是结点 e 的孩子?
- (7) 哪些是结点 e 的兄弟? 哪些是结点 f 的兄弟?
- (8) 结点 b 和 n 的层次号分别是什么?
- (9) 树的深度是多少?
- (10) 以结点 c 为根的子树深度是多少?

答: (1) 根结点: a

对应的树为:



(2) 叶子结点: d、m、n、j、k、f、l;

(3) 结点 g 的双亲: c

(4) 结点 g 的祖先: a、c

(5) 结点 g 的孩子: j、k

(6) 结点 e 的孩子 (子孙): i、m、n

(7) 结点 e 的兄弟: d。结点 f 的兄弟: g、h

(8) 结点 b 的层次=2, 结点 n 的层次=5

(9) 树的深度=5

(10) 以结点 c 为根的子树深度是 3

3、一棵度为2的树与一棵二叉树有何区别。(评分标准：共5分。第1点4分，第2点1分!)

答：度为2的树有两个分支，但分支没有左右之分；一棵二叉树也有两个分支，但有左右之分，左右子树不能交换。

4、试分别画出具有3个结点的树和二叉树的所有不同形态？(评分标准：共5分。第1点4分，第2点1分!)

答：(1)。(2)；

5、已知用一维数组存放的一棵完全二叉树：ABCDEFGHIJKL，写出该二叉树的先序、中序和后序遍历序列。(评分标准：共5分。第1点4分，第2点1分!)

答：

先序序列：ABDHIEJKCFLG

中序序列：HDIBJEKALFCG

后序序列：HIDJKEBLFGCA

6、找出所有满足下列条件的二叉树：(评分标准：共5分。第1点4分，第2点1分!)

(1) 它们在先序遍历和中序遍历时，得到的遍历序列相同；

(2) 它们在后序遍历和中序遍历时，得到的遍历序列相同；

(3) 它们在先序遍历和后序遍历时，得到的遍历序列相同；

答：

(1) 先序序列和中序序列相同的二叉树为：空树或者任一结点均无左孩子的非空二叉树；

(2) 中序序列和后序序列相同的二叉树为：空树或者任一结点均无右孩子的非空二叉树；

(3) 先序序列和后序序列相同的二叉树为：空树或仅有一个结点的二叉树。

7、假设一棵二叉树的先序序列为EBADCFHGIKJ，中序序列为ABCDEFGHIJK，请写出该二叉树的后序遍历序列。(评分标准：共5分。第1点4分，第2点1分!)

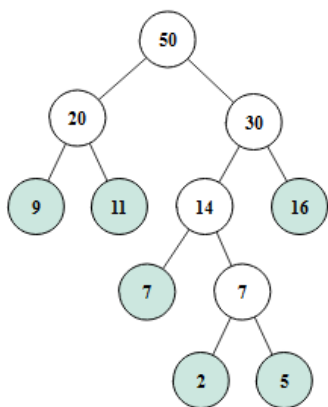
答：后序序列：ACDBGJKIHFE。

8、假设一棵二叉树的后序序列为DCEGBFHKJIA，中序序列为DCBGEAHFIJK，请写出该二叉树的先序遍历序列。(评分标准：共5分。第1点4分，第2点1分!)

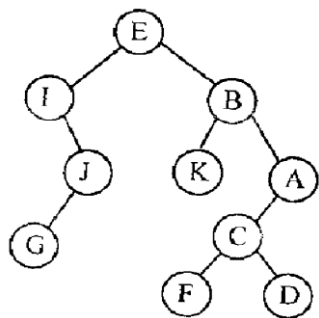
答：先序序列：ABCDGEIHFJK。

9、给定一组权值(5, 9, 11, 2, 7, 16)，试设计相应的哈夫曼树。(评分标准：共5分。第1点4分，第2点1分!)

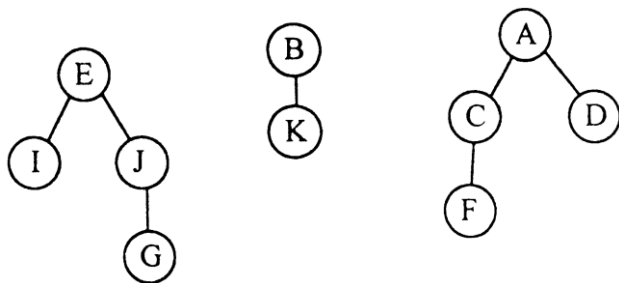
答：构造而成的哈夫曼树如下：



10、请画出与下列二叉树对应的森林。(评分标准：共5分。第1点4分，第2点1分!)



答：对应的森林为：



11、设完全二叉树的顺序存储结构中存储数据 ABCDE，要求给出该二叉树的链式存储结构并给出该二叉树的前序、中序和后序遍历序列。(评分标准：共5分。第1点4分，第2点1分!)

答：链式存储结构略。

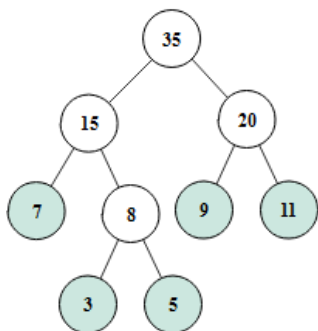
前序：ABDEC；

中序：DBEAC；

后序：DEBCA。

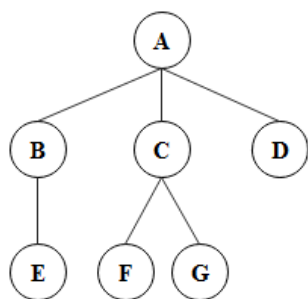
12、设给定一个权值集合 $W=(3, 5, 7, 9, 11)$ ，要求根据给定的权值集合构造一棵哈夫曼树并计算哈夫曼树的带权路径长度 WPL。(评分标准：共5分。第1点4分，第2点1分!)

答：(1) 哈夫曼树：



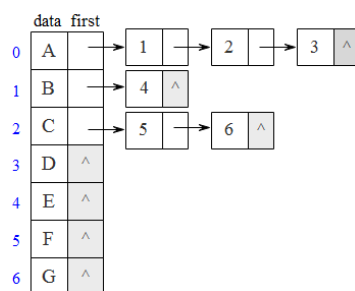
(2) WPL=78;

- 13、设一棵树 T 中边的集合为 $\{(A, B), (A, C), (A, D), (B, E), (C, F), (C, G)\}$ ，要求用孩子兄弟表示法（二叉链表）表示出该树的存储结构并将该树转化成对应的二叉树。（评分标准：共5分。第1点4分，第2点1分!）



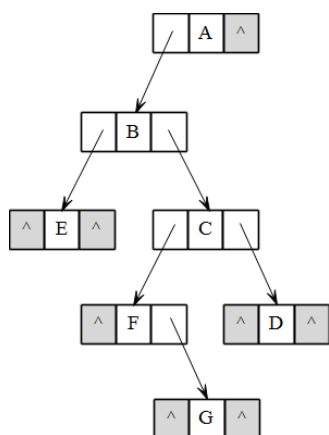
答：

[a]树

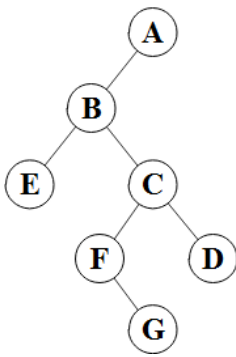


[b]链式存储结构

(1) 孩子兄弟表示法（二叉链表）表示的树的存储结构：



(2) 树对应的二叉树：



14、设某棵二叉树的中序遍历序列为 DBEAC，前序遍历序列为 ABDEC，要求给出该二叉树的后序遍历序列。（评分标准：共5分。第1点4分，第2点1分!）

答：DEBCA；

得分

六、编程题（X题，每题X分，共X分）

评分标准：每小题回答正确得10分，不完全正确则按得分点给分。

1、设二叉链树的类型定义如下，试写出求该二叉树叶子结点数的算法。实现算法的函数原型为：

Status **CountLeaves**(BinTree &root, int &n); （评分标准：第1点4分，第2点1分!）

```
typedef int Elemtype;
```

```
typedef struct node {
```

```
    Elemtype data;
```

```
    struct node *lchild, *rchild;
```

```
} BinNode, *BinTree;
```

答：Status **CountLeaves**(BinTree &root, int &n) { //n is the number of leaves

```
    if(T) {
```

```
        if( !(T->lchild) && !(T->rchild))
```

```
            n++;
```

```
        CountLeaves (T->lchild, n);
```

```
        CountLeaves (T->rchild, n);
```

```
    }
```

```
}
```

2、设计在链式存储结构上建立一棵二叉树的算法。实现算法的函数原型为： void **createbitree**(bitree *&bt); （评分标准：第1点4分，第2点1分!）

答：typedef char datatype;

```
typedef struct node {
```

```
    datatype data;
```

```
    struct node *lchild, *rchild;
```

```
} bitree;
```

```
void createbitree(bitree *&bt) {  
    char ch;  
    scanf("%c",&ch);  
    if(ch=='#') {  
        bt=0;  
        return;  
    }  
  
    bt=(bitree*)malloc(sizeof(bitree));  
    bt->data=ch;  
  
    createbitree(bt->lchild);  
    createbitree(bt->rchild);  
}
```

3、设计判断两个二叉树是否相同的算法。实现算法的函数原型为：int **judgebitree**(bitree *bt1,bitree *bt2); (评分标准：第1点4分，第2点1分!)

答：typedef struct node {
 datatype data;
 struct node *lchild,*rchild;
} bitree;

```
int judgebitree(bitree *bt1,bitree *bt2) {  
    if (bt1==0 && bt2==0)  
        return(1);  
    else if(bt1==0 || bt2==0 ||bt1->data!=bt2->data)  
        return(0);  
    else  
        return (judgebitree(bt1->lchild,bt2->lchild)*judgebitree(bt1->rchild,bt2->rchild));  
}
```

4、设计一个在链式存储结构上统计二叉树中结点个数的算法。实现算法的函数原型为：void **countnode**(bitree *bt,int &count); (评分标准：第1点4分，第2点1分!)

答：void countnode(bitree *bt,int &count) {
 if(bt!=0) {
 count++;
 countnode(bt->lchild,count);
 countnode(bt->rchild,count);
 }
}

5、设计计算二叉树中所有结点值之和的算法。实现算法的函数原型为：void **sum**(bitree *bt, int &s); (评分标准：第1点4分，第2点1分!)

答：void **sum**(bitree *bt, int &s) {
 if(bt!=0) {
 s=s+bt->data;

 sum(bt->lchild,s);
 sum(bt->rchild,s);
 }
}