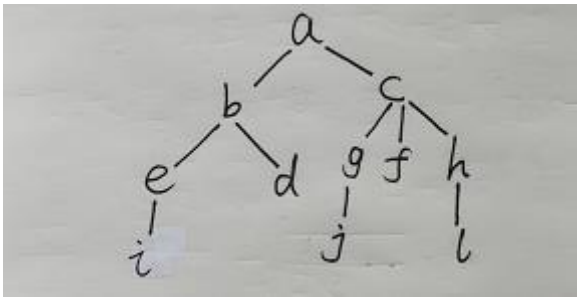
1、假设在树中，结点 x 是结点 y 的双亲时，用(x, y)来表示树边。已知一棵树的树边集合为{(e, i), (b, e), (b, d), (a, b), (g, j), (c, g), (c, f), (h, l), (c, h), (a, c)}，用树型表示法表示该树，并回答下列问题：

1）哪个是根结点?哪些是叶子结点?哪个是 g 的双亲?哪些是 g 的祖先?哪些是 g 的孩子?那些是 e 的子孙?哪些是 e 的兄弟?哪些是 f 的兄弟?

2）b 和 n 的层次各是多少?树的深度是多少?以结点 c 为根的子树的深度是多少?

解：画图可知该树如下图所示：



所以（1）根结点：a

叶子结点：i, d, j, f, l

g 的双亲：c

g 的祖先：c, a

g 的孩子：j

e 的子孙：i

e 的兄弟：d

f 的兄弟：g, h

（2）b 的层次：2

h 的层次：3

树的深度：4

以结点 c 为根的子树的深度：3

2、一棵深度为 h 的满 k 叉树有如下性质： 第 h 层上的结点都是叶子结点，其余各层上每个结点都有 k 棵非空子树。 如果按层次顺序(同层自左至右)从 1 开始对全部结点编号，问：

1） 各层的结点数是多少?

2） 编号为 i 的结点的双亲结点(若存在)的编号是多少?

3）编号为 i 的结点的第 j 个孩子结点(若存在)的编号是多少?

4）编号为 i 的结点的有右兄弟的条件是什么? 其右兄弟的编号是多少?

解：（1）第 i 层的结点数为：k^{i-1}个

（2）编号为：$\frac{i+k-2}{k}$（向下取整）

（3）$(i-1)k+j+1$

（4）$i \neq (m-1)k+4$

3、设有如图 1 所示的二叉树。

1）分别用顺序存储方法和链接存储方法画出该二叉树的存储结构。

2） 写出该二叉树的先序、中序、后序遍历序列。

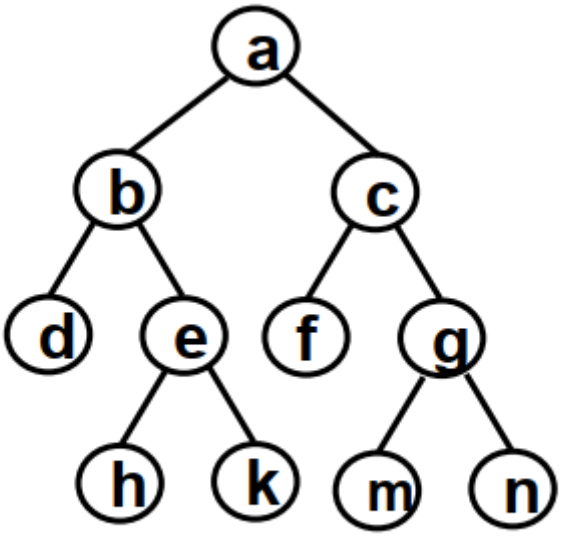
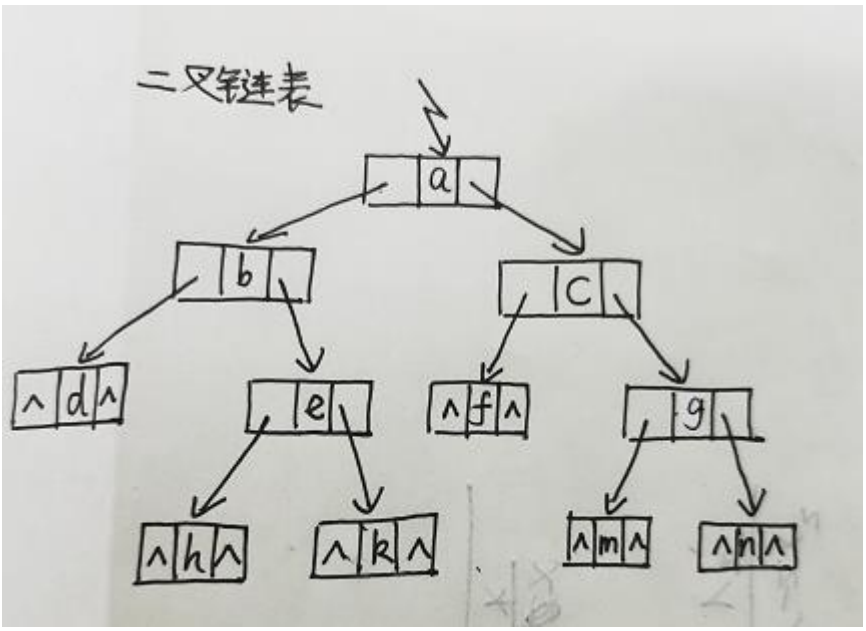


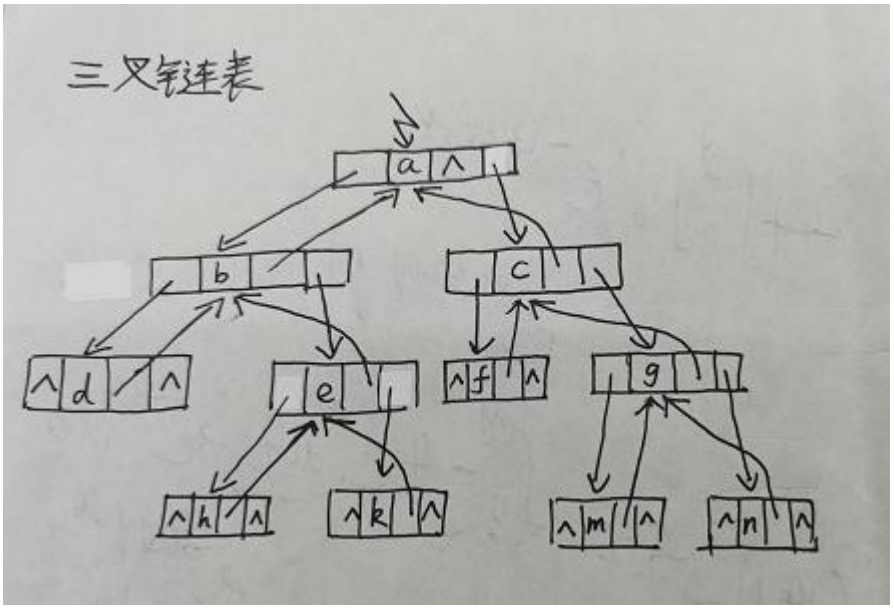
图1 二叉树

解：（1）对于顺序存储：

a	b	c	d	e	f	g	^	^	h	k	^	^	m	n

对于链接存储：





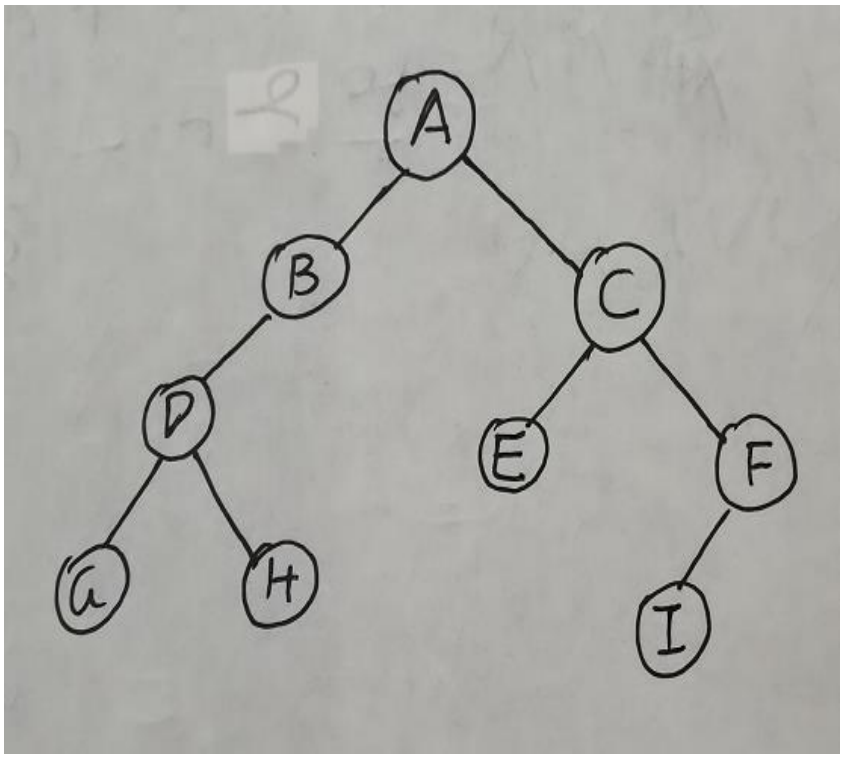
（2）先序遍历：a b d e h k c f g m n

中序遍历：d b h e k a f c m g n

后序遍历：d h k e b f m n g c a

4、已知一棵二叉树的先序遍历序列和中序遍历序列分别为 ABDGHCEFI 和 GDHBAECIF，请画出这棵二叉树，然后给出该树的后序遍历序列。

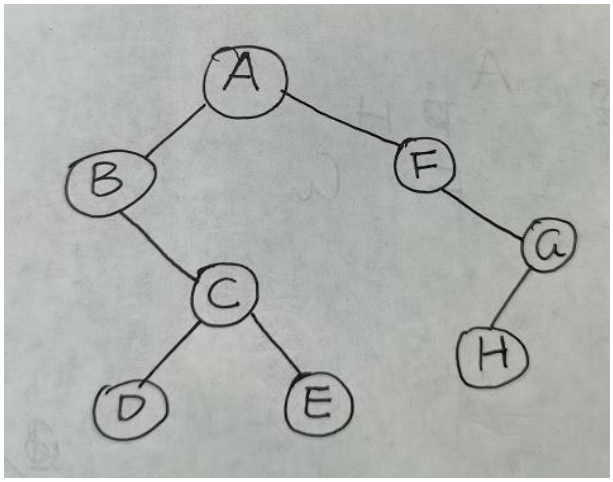
解：该二叉树如下图所示：



所以后序遍历为：G H D B E I F C A

5、设一棵二叉树的中序遍历序列和后序遍历序列分别为 BDCEAFHG 和 DECBHGFA ，请画出这棵二叉树，然后给出该树的先序序列。

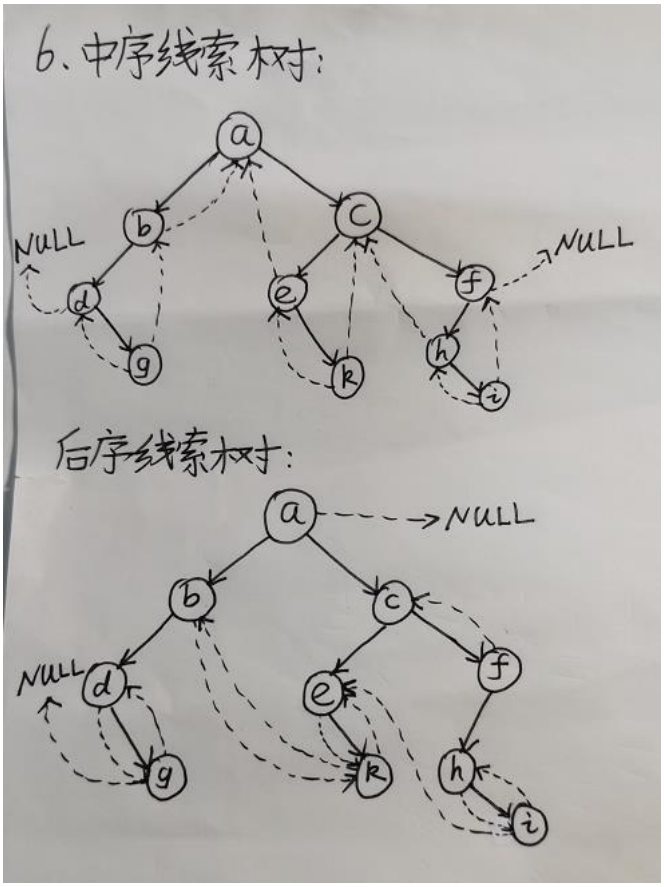
解：该二叉树如下图所示：



所以先序遍历为：A B C D E F G H

6、已知一棵二叉树的中序遍历序列和后序遍历序列分别为 dgbaekchif 和 gdbkeihfca，请画出这棵二叉树对应的中序线索树和后序线索树。

解：



7、设有一棵树，如图 2 所示。

1）请分别用双亲表示法、孩子表示法、孩子兄弟表示法给出该树的存储结构。

2）请给出该树的先序遍历序列和后序遍历序列。

3）请将这棵树转换成二叉树。

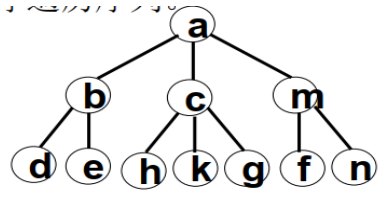
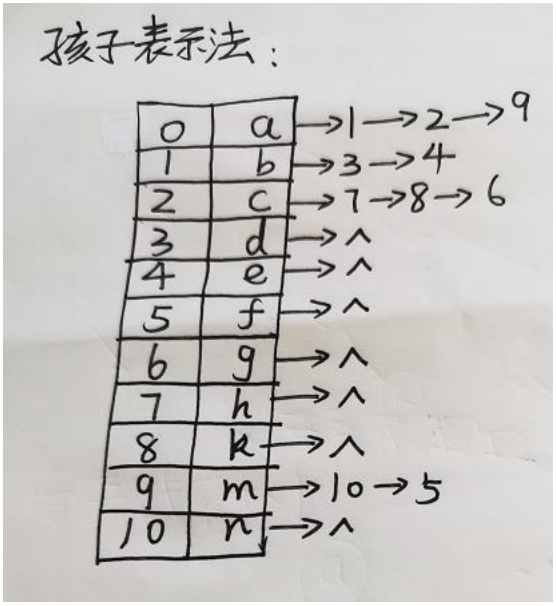


图2 一般的树

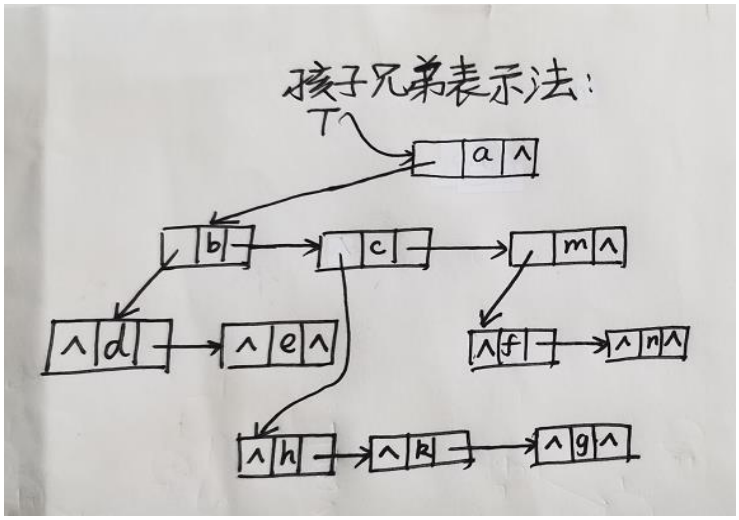
解：（1）双亲表示法：

下标	info	parent
0	a	-1
1	b	0
2	c	0
3	d	1
4	e	1
5	f	9
6	g	2
7	h	2
8	k	2
9	m	0
10	n	9

孩子表示法：



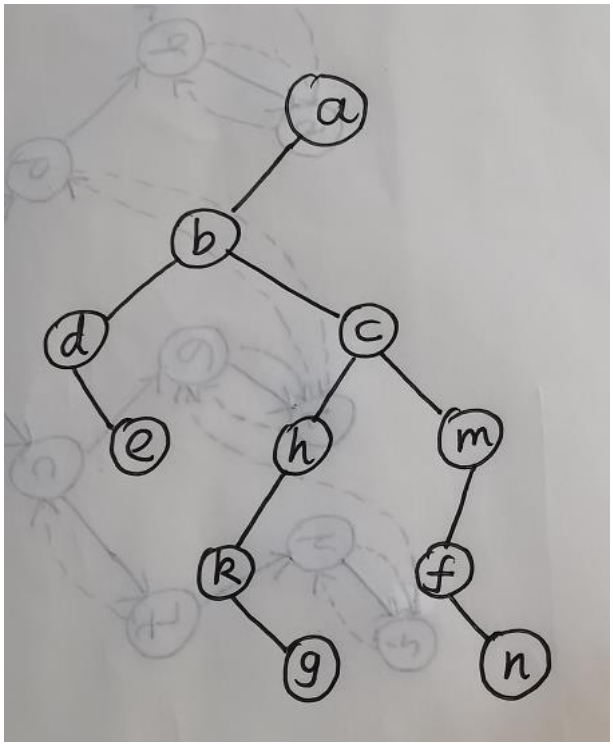
孩子兄弟表示法：



（2）先序遍历：a b d e c h k g m f n

后序遍历：d e b h k g c f n m a

（3）转换图如下所示：



8、设二叉树 t 的存储结构如图 3 所示。其中 t 为树根结点的指针，Left 和 Right 分别为结点的左、右孩子指针域， data 为结点的数据域，请完成下列各题：

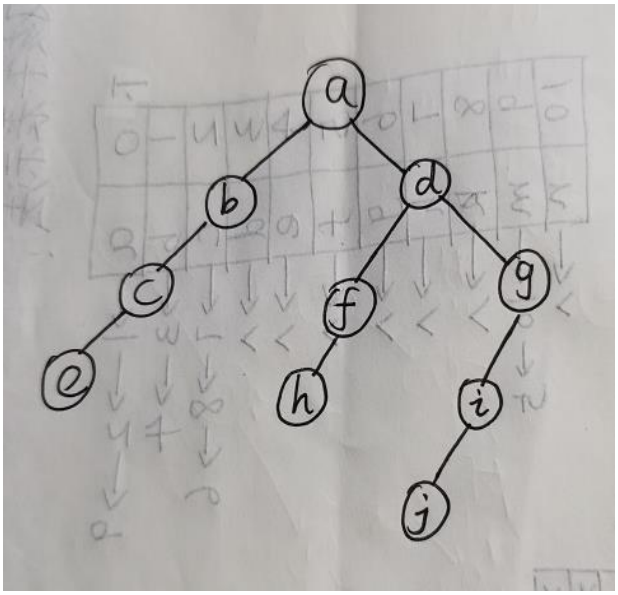
1）画出二叉树 t 的逻辑结构

2）写出按前序、中序和后序遍历二叉树 t 所得到的结点序列。

	1	2	3	4	5	6	7	8	9	10
Left	0	0	2	3	7	5	8	0	10	1
Data	j	h	f	d	b	a	c	e	g	i
Right	0	0	0	9	4	0	0	0	0	0

图3

解：（1）



（2）前序：a b c e d f h g i j

中序：e c b h f d j i g a

后序：e c h f j i g d b a

9、表 1 中 m, n 分别是一棵二叉树中两个结点，行号 i=1, 2, 3, 4 分别表示四种 m, n 的相对关系，列号 j=1, 2, 3 分别表示在前序、中序和后序遍历中 m, n 之间的先后次序关系，要求 i, j 所表示的关系能够同时发生的方格内打“✓”。

	i \ j	前序遍历n先被访问	中序遍历n先被访问	后序遍历n先被访问
1	n在m的左边			
2	n在m的右边			
3	n是m的祖先			
4	n是m的儿子			

解：

	i \ j	前序遍历n先被访问	中序遍历n先被访问	后序遍历n先被访问
1	n在m的左边	✓	✓	✓
2	n在m的右边			
3	n是m的祖先	✓		
4	n是m的儿子			✓

10、假设二叉树采用二叉链表存储，编写一个后序遍历二叉树的非递归算法。

解：

```
struct binTreeNode{
    value_type data;
    binTreeNode* left,right,parent;
};
void posOrder(binTreeNode* root){
    if(root!=NULL){
        posOrder(root->left);
        posOrder(root->right);
        read(root);
    }
}
```

11、在二叉树中查找值为 X 的结点，设计打印值为 X 的结点的双亲的算法。

解：

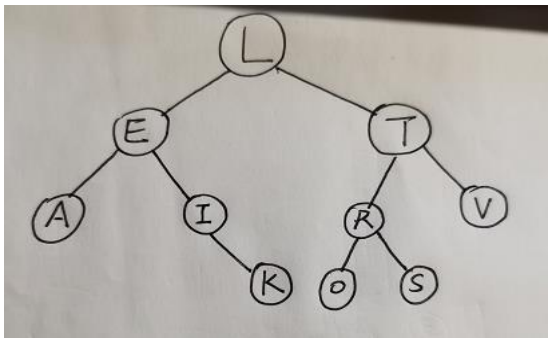
```
struct binTreeNode{
    value_type data;
    binTreeNode* left,right,parent;
};
binTreeNode* search(BinTreeNode* root const value_type value)
{
    if(root->data==value){
        cout<<root->parent->data<<endl;
        return root;
    }
    binTreeNode* ans;
    ans=search(root->left,value);
    if(ans!=NULL){
        return ans;
    }
    ans=search(root->right,value);
    if(ans!=NULL){
        return ans;
    }
    return NULL;
}
```

12、以下列顺序插入数据元素，并用边建立边平衡的方式建立 AVL 二叉排序树。

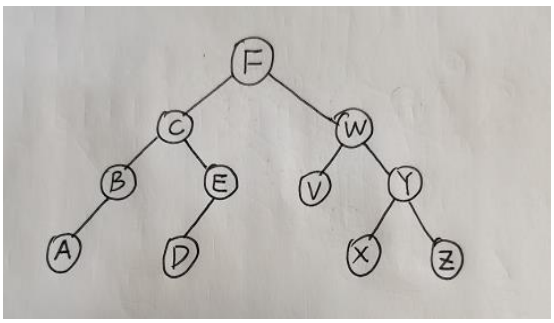
1）A,V,L,T,R,E,I,S,O,K

2）A,Z,B,Y,C,X,D,W,E,V,F

解：（1）

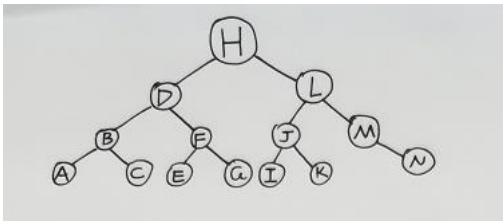


（2）

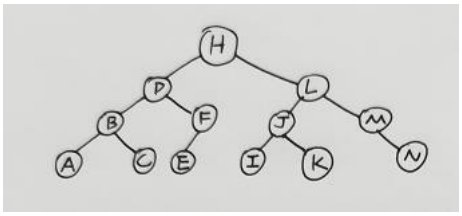


13、以 A-N 为关键字，建立 AVL 树，并对建立好的树，图示依次删除 G, D, N 结点。

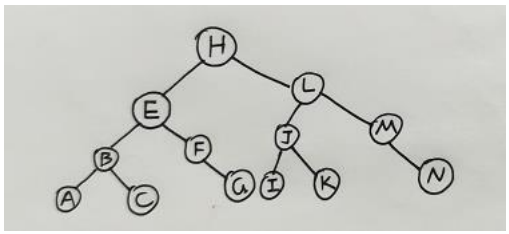
解：原始的 AVL 树：



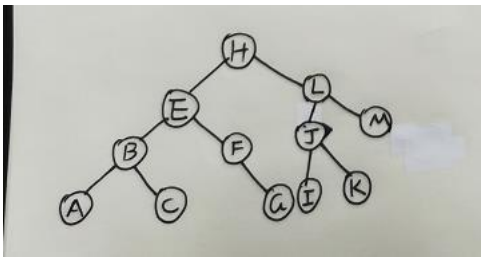
删除 G 之后：



删除 D 之后：

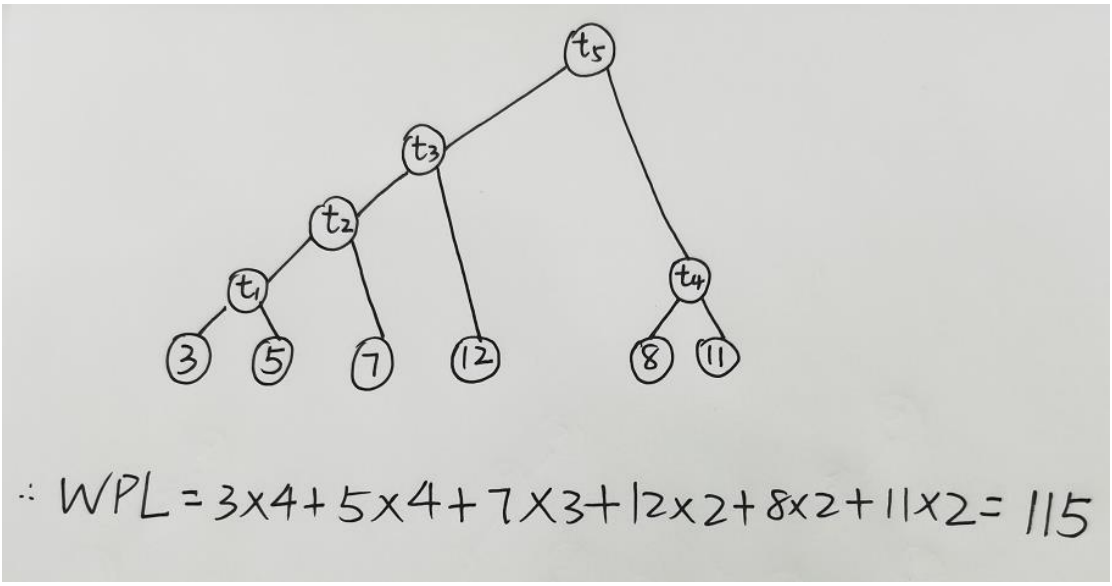


删除 N 之后：



16、设给定权值集合 w={3,5,7,8,11,12}，请构造关于 w 的一棵 huffman 树，并求其加权路径长度 WPL。

解：

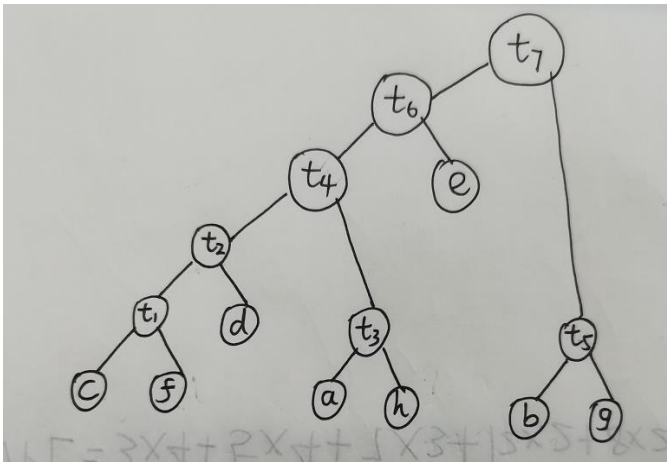


17、假设用于通信的电文是由字符集{a,b,c,d,e,f,g,h}中的字符构成，这 8 个字符在电文中出现的概率分别为{0.07,0.19,0.02,0.06,0.32,0.03,0.21,0.10}。

1)请画出对应的 huffman 树(按左子树根结点的权小于等于右子树根结点的权的次序构造)。

2) 求出每个字符的 huffman 编码。

解：(1)



（2）a:0010

b:10

c:00000

d:0001

e:01

f:00001

g:11

h:0011