**一：数据结构与算法基础部分**

1：线性表的头结点 ①可能有也可能没有

②存在时不能被删除

③存在时一定是线性表的第一个几点

2：已知链表中的一个结点a，在a的后面加上一个结点b。b→(a+1) a→b

3 : 平衡二叉树： ①左子树与右子树深度差不超过1

②平衡二叉树的子树也是平衡二叉树

4：度：结点有多少个子结点，则该结点的度就是多少

5：空树的根结点数为0，其余的树的根结点数都为1

6：**题型**：若一个树有2个度为3的结点，4个度为2的结点，则一共有多少个叶子结点（注意问的是总结点数还是叶子结点数！）

结合概念4,5，画图

总结点数 13 叶子结点数 12

7： ①先序序列 ：根，左，右

二叉树的遍历 ②中序序列 ：左，根，右

③后序序列 ：左，右，根

先、中、后 指的是根结点的位置

8：lg x = log10 x ; ln x = loge x ; lb x = log2 x

9：ASL（平均查找长度）：数组中A被查找的概率 X 查找A需要进行多少次对比 = A的查找长度 ASL = A+B+C…

10：前缀编码：在前缀编码中，所有的字符都不能是另外一个字符的前缀编码，哈夫曼编码就是一种前缀编码

11：哈夫曼树：一种二叉树

12：**题型**：已知权值数组[1,3,4,9,6]，求对应的哈夫曼树。

解题步骤：①将每个权值看成一颗只有一个跟结点的树，②找出权值最小的两颗树，组合成一颗新树（哪个权值在左，哪个权值在右无限制），并放回数组中。③重复步骤②直到只剩一棵树

**13：线索二叉树**

定义：按照先序、中序或者后序，用虚线将节点之间的顺序标明。后继：当前结点后一个结点的意思，前驱：当前结点前一个结点的意思。

考点：中序线索二叉树容易找到前驱和后继；先序线索二叉树容易找到后继；后序线索二叉树容易找到前驱；

微观定义：一个节点，可以分为

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | E |

五部分。A、E指向其他结点，C中存的是本结点的数据，B和D的值域为0、1，B = 0 表示A指向左子树 B = 1 表示A指向前驱 D = 0 表示E指向右子树，D = 1表示E指向后继。

连接时，实线表示父子关系，虚线表示前驱后继关系

14：二叉排序树：又叫二叉查找树，二叉查找树的权值 左子女<=根结点<=右子女

15：图：①由点和边组成，点非空且有穷

②无向图：边都没有方向，用（a,b）表示边

③有向图：边都有方向，用<a,b>表示边

④<a,b>是有向边，也称为弧，a是弧尾，b是弧头

⑤完全图：当前点下 边的数目为最大

⑥每条边都可以有权重，表示从a到b的难度，所有边都有权重的图就是网

⑦每个顶点上连接了多少条边这个点的度就是多少。有向图还分了出度和入度。

⑧任意点互通的无向图被称为连通图。任意点互通的有向图被称为强连通图

⑨删除连通图的一些边，可以生成树，剩余边的权重最小时，就是最小生成树。