第1章绪论练习题答案

**判断题**

1. 数据元素是数据的最小单位。( × )

2. 记录是数据处理的最小单位。 (× )

3. 数据的逻辑结构是指数据的各数据项之间的逻辑关系。(× )

4．算法的优劣与算法描述语言无关，但与所用计算机有关。(× )

5．健壮的算法不会因非法的输入数据而出现莫名其妙的状态。(√ )

6．数据的物理结构是指数据在计算机内的实际存储形式。(√ )

7. 数据结构的抽象操作的定义与具体实现有关。(× )

**第2章线性表练习题答案**

**一、填空**

1. 向一个长度为n的向量的第i个元素(1≤i≤n+1)之前插入一个元素时，需向后移动 n-i+1 个元素。

2. 向一个长度为n的向量中删除第i个元素(1≤i≤n)时，需向前移动 n-i 个元素。

3. 在顺序表中访问任意一结点的时间复杂度均为 O(1) ，因此，顺序表也称为 随机存取 的数据结构。

4. 在单链表中，除了首元结点外，任一结点的存储位置由 其直接前驱结点的链域的值 指示。

5．对于一个具有n个结点的单链表，在已知的结点\*p后插入一个新结点的时间复杂度为 O(1) ，在给定值为x的结点后插入一个新结点的时间复杂度为 O(n) 。

**二、判断正误**

（ × ）1. 链表的每个结点中都恰好包含一个指针。

答：错误。链表中的结点可含多个指针域，分别存放多个指针。例如，双向链表中的结点可以含有两个指针域，分别存放指向其直接前趋和直接后继结点的指针。

（ × ）2. 链表的物理存储结构具有同链表一样的顺序。错，链表的存储结构特点是无序，而链表的示意图有序。

（ × ）3. 链表的删除算法很简单，因为当删除链中某个结点后，计算机会自动地将后续的各个单元向前移动。错，链表的结点不会移动，只是指针内容改变。

（ × ）4. 线性表的每个结点只能是一个简单类型，而链表的每个结点可以是一个复杂类型。

错，混淆了逻辑结构与物理结构，链表也是线性表！且即使是顺序表，也能存放记录型数据。

（ × ）5. 顺序表结构适宜于进行顺序存取，而链表适宜于进行随机存取。

错，正好说反了。顺序表才适合随机存取，链表恰恰适于“顺藤摸瓜”

（ × ）6. 顺序存储方式的优点是存储密度大，且插入、删除运算效率高。

错，前一半正确，但后一半说法错误，那是链式存储的优点。顺序存储方式插入、删除运算效率较低，在表长为n的顺序表中，插入和删除一个数据元素，平均需移动表长一半个数的数据元素。

（ × ）7. 线性表在物理存储空间中也一定是连续的。

错，线性表有两种存储方式，顺序存储和链式存储。后者不要求连续存放。

（ × ）8. 线性表在顺序存储时，逻辑上相邻的元素未必在存储的物理位置次序上相邻。

错误。线性表有两种存储方式，在顺序存储时，逻辑上相邻的元素在存储的物理位置次序上也相邻。

（ × ）9. 顺序存储方式只能用于存储线性结构。

错误。顺序存储方式不仅能用于存储线性结构，还可以用来存放非线性结构，例如完全二叉树是属于非线性结构，但其最佳存储方式是顺序存储方式。（后一节介绍）

（ × ）10. 线性表的逻辑顺序与存储顺序总是一致的。

错，理由同7。链式存储就无需一致。

**三、简答题**

1.试比较顺序存储结构和链式存储结构的优缺点。在什么情况下用顺序表比链表好？

**答：① 顺序存储时，相邻数据元素的存放地址也相邻（逻辑与物理统一）；要求内存中可用存储单元的地址必须是连续的。**

**优点：存储密度大（＝1？），存储空间利用率高。缺点：插入或删除元素时不方便。**

**②链式存储时，相邻数据元素可随意存放，但所占存储空间分两部分，一部分存放结点值，另一部分存放表示结点间关系的指针**

**优点：插入或删除元素时很方便，使用灵活。缺点：存储密度小（<1），存储空间利用率低。**

**顺序表适宜于做查找这样的静态操作；链表宜于做插入、删除这样的动态操作。**

**若线性表的长度变化不大，且其主要操作是查找，则采用顺序表；**

**若线性表的长度变化较大，且其主要操作是插入、删除操作，则采用链表。**

2.描述以下三个概念的区别：头指针、头结点、首元结点（第一个元素结点）。在单链表中设置头结点的作用是什么？

**答：首元结点是指链表中存储线性表中第一个数据元素a1的结点。为了操作方便，通常在链表的首元结点之前附设一个结点，称为头结点，该结点的数据域中不存储线性表的数据元素，其作用是为了对链表进行操作时，可以对空表、非空表的情况以及对首元结点进行统一处理。头指针是指向链表中第一个结点（或为头结点或为首元结点）的指针。若链表中附设头结点，则不管线性表是否为空表，头指针均不为空。否则表示空表的链表的头指针为空。这三个概念对单链表、双向链表和循环链表均适用。是否设置头结点，是不同的存储结构表示同一逻辑结构的问题。**

**头结点**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **head** | **🡪** | **data** | **link** |

**头指针 首元结点**

简而言之，

**头指针是指向链表中第一个结点（或为头结点或为首元结点）的指针；**

**头结点是在链表的首元结点之前附设的一个结点；**

**首元素结点是指链表中存储线性表中第一个数据元素a1的结点。**

**第3章栈和队列练习题答案**

一、填空题

1. 向量、栈和队列都是 线性 结构，可以在向量的 任何 位置插入和删除元素；对于栈只能在 栈顶 插入和删除元素；对于队列只能在 队尾 插入和 队首 删除元素。

2. 栈是一种特殊的线性表，允许插入和删除运算的一端称为 栈顶 。不允许插入和删除运算的一端称为 栈底 。

3. 队列 是被限定为只能在表的一端进行插入运算，在表的另一端进行删除运算的线性表。

4. 在具有n个单元的循环队列中，队满时共有 n-1 个元素。

二、判断正误

（ √ ）1. 栈是一种对所有插入、删除操作限于在表的一端进行的线性表，是一种后进先出型结构。

（ √ ）2. 对于不同的使用者，一个表结构既可以是栈，也可以是队列，也可以是线性表。

正确，都是线性逻辑结构，栈和队列其实是特殊的线性表，对运算的定义略有不同而已。

（ × ）3. 栈和队列是一种非线性数据结构。

错，他们都是线性逻辑结构，栈和队列其实是特殊的线性表，对运算的定义略有不同而已。

（ √ ）4. 栈和队列的存储方式既可是顺序方式，也可是链接方式。

（ √ ）5. 两个栈共享一片连续内存空间时，为提高内存利用率，减少溢出机会，应把两个栈的栈底分别设在这片内存空间的两端。

（ × ）6. 队是一种插入与删除操作分别在表的两端进行的线性表，是一种先进后出型结构。

错，后半句不对。

（ × ）7. 一个栈的输入序列是12345，则栈的输出序列不可能是12345。

错，有可能。

三、应用题

利用算法3.10对算术表达式2\*(1+3)进行求值，给出其求值的具体过程。

参考教材表3.2。

表3.2 算术表达式2\*(1+3)的求值过程

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 步 骤 | OPTR栈 | OPND栈 | 读 入 字 符 | 主 要 操 作 |
| 1 | # |  | 2\*(1+3)# | Push(OPND,‘2’) |
| 2 | # | 2 | \*(1+3)# | Push(OPTR,‘\*’) |
| 3 | #\* | 2 | (1+3)# | Push(OPTR,‘(’) |
| 4 | #\*( | 2 | 1+3)# | Push(OPND，‘1’) |
| 5 | #\*( | 2 1 | +3)# | Push(OPTR，‘+’) |
| 6 | #\*(+ | 2 1 | 3)# | Push(OPND，‘3’) |
| 7 | #\*(+ | 2 1 3 | )# | Push(OPND，Operate(‘1’，‘+’，‘3’)) |
| 8 | #\*( | 2 4 | )# | Pop(OPTR){消去一对括号} |
| 9 | #\* | 2 4 | # | Push(OPND，Operate(‘2’，‘\*’，‘4’)) |
| 10 | # | 8 | # | return(GetTop(OPND)) |

**第4章串、数组和广义表练习题答案**

**填空题**

1. 不包含任何字符（长度为0）的串 称为空串。

2. 由一个或多个空格（仅由空格符）组成的串 称为空白串。

3. 设S=“A;/document/Mary.doc”，则strlen(s)= 20 , “/”的字符定位的位置为 3 。

4. 子串的定位运算称为串的模式匹配； 被匹配的主串 称为目标串， 子串 称为模式。

5. 设目标T=”abccdcdccbaa”，模式P=“cdcc”，则第 6 次匹配成功。

6. 若n为主串长，m为子串长，则串的古典（朴素）匹配算法最坏的情况下需要比较字符的总次数为 (n-m+1)\*m 。

7.设数组a[1…60, 1…70]的基地址为2048，每个元素占2个存储单元，若以列序为主序顺序存储，则元素a[32,58]的存储地址为 8950

。

**答：不考虑0行0列，利用列优先公式，2048+（57\*60+31）\*2=8950**

8.求下列广义表操作的结果：

（1） GetHead【((a,b),(c,d))】=== (a, b) ; //头元素不必加括号

（2） GetHead【GetTail【((a,b),(c,d))】】=== (c,d) ;

（3） GetHead【GetTail【GetHead【((a,b),(c,d))】】】=== b ;

（4） GetTail【GetHead【GetTail【((a,b),(c,d))】】】=== （d） ;

**第5章树和二叉树练习题答案**

**一、下面是有关二叉树的叙述，请判断正误**

（ √ ）1. 若二叉树用二叉链表作存贮结构，则在n个结点的二叉树链表中只有n—1个非空指针域。

（ × ）2.二叉树中每个结点的两棵子树的高度差等于1。

（ √ ）3.二叉树中每个结点的两棵子树是有序的。

（ × ）4.二叉树中每个结点有两棵非空子树或有两棵空子树。

（ × ）5.二叉树中每个结点的关键字值大于其左非空子树（若存在的话）所有结点的关键字值，且小于其右非空子树（若存在的话）所有结点的关键字值。 （应当是二叉排序树的特点）

（ × ）6.二叉树中所有结点个数是2k-1-1，其中k是树的深度。（应2k-1）

（ × ）7.二叉树中所有结点，如果不存在非空左子树，则不存在非空右子树。

（ × ）8.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2i—1个结点。（应2i-1）

（ √ ）9.用二叉链表法存储包含n个结点的二叉树，结点的2n个指针区域中有n+1个为空指针。

（正确。用二叉链表存储包含n个结点的二叉树，结点共有2n个链域。由于二叉树中，除根结点外，每一个结点有且仅有一个双亲，所以只有n-1个结点的链域存放指向非空子女结点的指针，还有n+1个空指针。）即有后继链接的指针仅n-1个。

（ √ ）10.具有12个结点的完全二叉树有5个度为2的结点。

**二、填空**

1． 由３个结点所构成的二叉树有 5 种形态。

2. 一棵深度为6的满二叉树有 n1+n2=0+ n2= n0-1=31 个分支结点和 26-1 =32 个叶子。

注：满二叉树没有度为1的结点，所以分支结点数就是二度结点数。

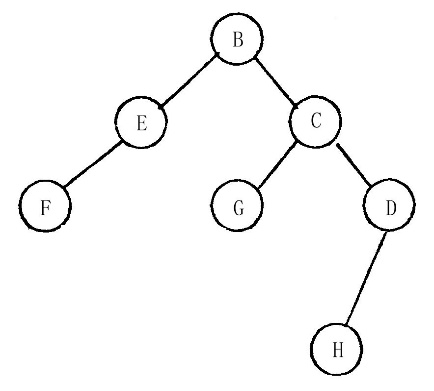
3． 一棵具有２５７个结点的完全二叉树，它的深度为 9 。

（ 注：用⎣ log2(n) ⎦+1= ⎣ 8.xx ⎦+1=9

1. 设一棵完全二叉树有700个结点，则共有 350 个叶子结点。

5. 设一棵完全二叉树具有1000个结点，则此完全二叉树有 500 个叶子结点，有 499 个度为2的结点，有 1 个结点只有非空左子树，有 0 个结点只有非空右子树。

答：最快方法：用叶子数＝[n/2]＝500 ，n2=n0-1=499。 另外，最后一结点为2i属于左叶子，右叶子是空的，所以有1个非空左子树。完全二叉树的特点决定不可能有左空右不空的情况，所以非空右子树数＝0.

6. 二叉树的基本组成部分是：根（N）、左子树（L）和右子树（R）。因而二叉树的遍历次序有六种。最常用的是三种：前序法（即按N L R次序），后序法（即按 L R N 次序）和中序法（也称对称序法，即按L N R次序）。这三种方法相互之间有关联。若已知一棵二叉树的前序序列是BEFCGDH，中序序列是FEBGCHD，则它的后序序列必是 F E G H D C B 。 解：法1：先由已知条件画图，再后序遍历得到结果；

法2：不画图也能快速得出后序序列，只要找到根的位置特征。由前序先确定root，由中序先确定左子树。例如，前序遍历BEFCGDH中，根结点在最前面，是B；则后序遍历中B一定在最后面。

法3：递归计算。如B在前序序列中第一，中序中在中间（可知左右子树上有哪些元素），则在后序中必为最后。如法对B的左右子树同样处理，则问题得解。

7.用5个权值{3, 2, 4, 5, 1}构造的哈夫曼（Huffman）树的带权路径长度是 33 。

解：先构造哈夫曼树，得到各叶子的路径长度之后便可求出WPL＝（4＋5＋3）×2＋（1＋2）×3=33

**(15)**

**(9) (6) （注：两个合并值先后不同会导致编码不同，即哈夫曼编码不唯一）**

**4 5 3 (3) （注：合并值应排在叶子值之后）**

**1 2**

**三、应用题**

1.设如下图所示的二叉树B的存储结构为二叉链表，root为根指针，结点结构为：（lchild,data,rchild）。其中lchild，rchild分别为指向左右孩子的指针，data为字符型，root为根指针，试回答下列问题：

C的结点类型定义如下：

struct node

{char data;

struct node \***lchild,** rchild;

};

C算法如下：

void traversal(struct node \*root)

{if (root)

{printf(“%c”, root->data);

traversal(root->lchild);

printf(“%c”, root->data);

traversal(root->rchild);

}

}

(1)对下列二叉树B，执行下列算法traversal(root)，试指出其输出结果；

(2)假定二叉树B共有n个结点，试分析算法traversal(root)的时间复杂度。（共8分）

A

B D

C F G

E

二叉树B

解：这是“先根再左再根再右”，比前序遍历多打印各结点一次，输出结果为：A B C C E E B A D F F D G G

特点：①每个结点肯定都会被打印两次；②但出现的顺序不同，其规律是：凡是有左子树的结点，必间隔左子树的全部结点后再重复出现；如A，B，D等结点。反之马上就会重复出现。如C，E，F，G等结点。

时间复杂度以访问结点的次数为主，精确值为2\*n，时间渐近度为O(n).

2.给定二叉树的两种遍历序列，分别是：

前序遍历序列：D，A，C，E，B，H，F，G，I； 中序遍历序列：D，C，B，E，H，A，G，I，F，

试画出二叉树B，并简述由任意二叉树B的前序遍历序列和中序遍历序列求二叉树B的思想方法。

解：方法是：由前序先确定root，由中序可确定root的左、右子树。然后由其左子树的元素集合和右子树的集合对应前序遍历序列中的元素集合，可继续确定root的左右孩子。将他们分别作为新的root，不断递归，则所有元素都将被唯一确定，问题得解。

**D**

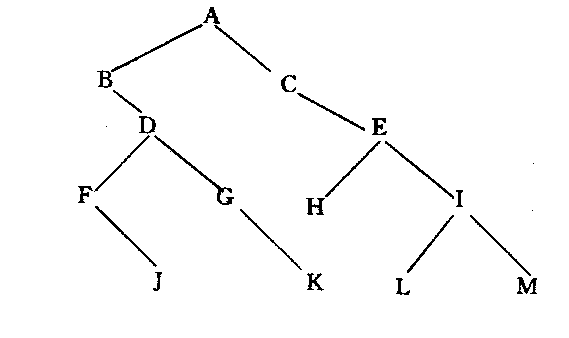
**A**

**C F**

**E G**

**B H I**

3.试写出如图所示的二叉树分别按先序、中序、后序遍历时得到的结点序列。



答：DLR：A B D F J G K C E H I L M

LDR: B F J D G K A C H E L I M

LRD：J F K G D B H L M I E C A

**第6章图练习题答案**

一、填空题

1. 图有 邻接矩阵 、 邻接表 等存储结构，遍历图有 深度优先遍历 、 广度优先遍历 等方法。

2. 有向图G用邻接表矩阵存储，其第i行的所有元素之和等于顶点i的 出度 。

3. n个顶点e条边的图，若采用邻接矩阵存储，则空间复杂度为 O(n2) 。

4. n个顶点e条边的图，若采用邻接表存储，则空间复杂度为 O(n+e) 。

5. 设有一稀疏图G，则G采用 邻接表 存储较省空间。

6. 一个有n个顶点的无向图最多有n(n-1)/2 条边。

7. 图的逆邻接表存储结构只适用于 有向 图。

8. 已知一个图的邻接矩阵表示，删除所有从第i个顶点出发的方法是 将邻接矩阵的第i行全部置0 。

9. 图的深度优先遍历序列 不是 惟一的。

10. n个顶点e条边的图采用邻接矩阵存储，深度优先遍历算法的时间复杂度为 O(n2) ；若采用邻接表存储时，该算法的时间复杂度为 O(n+e) 。

11. n个顶点e条边的图采用邻接矩阵存储，广度优先遍历算法的时间复杂度为 O(n2) ；若采用邻接表存储，该算法的时间复杂度为 O(n+e) 。

12. 图的BFS生成树的树高比DFS生成树的树高 小或相等 。

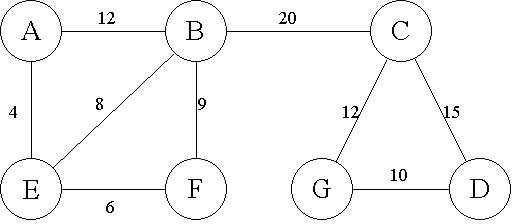
13. 若要求一个稀疏图G的最小生成树，最好用 克鲁斯卡尔(Kruskal) 算法来求解。

14. 若要求一个稠密图G的最小生成树，最好用 普里姆(Prim) 算法来求解。

15. 用Dijkstra算法求某一顶点到其余各顶点间的最短路径是按路径长度 递增 的次序来得到最短路径的。

16. 拓扑排序算法是通过重复选择具有 0 个前驱顶点的过程来完成的。

二、应用题

.给定下列网G

(1) 找出网G的最小生成树，画出其逻辑结构图；

(2 )用两种不同的表示法画出网G的存储结构图。

A B———————C

E————F G————D

解：

1. 最小生成树可直接画出，如右图所示。

2. 可用邻接矩阵和邻接表来描述（注意标注权值）：

邻接矩阵为：



邻接表为：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | → | b | 12 | → | e | 4 | ^ |  |  |  |  |  |  |
| b | → | a | 12 | → | c | 20 | → | e | 8 | → | f | 9 | ^ |
| c | → | b | 20 | → | d | 15 | → | g | 12 | ^ |  |  |  |
| d | → | c | 15 | → | g | 10 | ^ |  |  |  |  |  |  |
| e | → | a | 4 | → | b | 8 | → | f | 6 | ^ |  |  |  |
| f | → | b | 9 | → | e | 6 | ^ |  |  |  |  |  |  |
| g | → | c | 12 | → | d | 10 |  |  |  |  |  |  |  |

**第7章查找练习题答案**

**一、填空题**

1. 在数据的存放无规律而言的线性表中进行检索的最佳方法是 顺序查找（线性查找） 。

2. 假设在有序线性表a[20]上进行折半查找，则比较一次查找成功的结点数为1；比较两次查找成功的结点数为 2 ；比较四次查找成功的结点数为 8 ；平均查找长度为 3.7 。

解：显然，平均查找长度＝O（log2n）<5次（25）。但具体是多少次，则不应当按照公式

来计算（即（21×log221）/20＝4.6次并不正确！）。因为这是在假设n＝2h-1的情况下推导出来的公式。应当用穷举法罗列：

全部元素的查找次数为＝（1＋2×2＋4×3＋8×4＋5×5）＝74； ASL＝74/20=3.7 ！！！

3． 在各种查找方法中，平均查找长度与结点个数n无关的查找方法是 哈希查找 。

**二、简答题**

对分（折半）查找适不适合链表结构的序列，为什么？用二分查找的查找速度必然比线性查找的速度快，这种说法对吗？

答：不适合！虽然有序的单链表的结点是按从小到大（或从大到小）顺序排列，但因其存储结构为单链表，查找结点时只能从头指针开始逐步搜索，故不能进行折半查找。

二分查找的速度在一般情况下是快些，但在特殊情况下未必快。例如所查数据位于首位时，则线性查找快；而二分查找则慢得多。

**第8章排序练习题答案**

**填空题**

1. 大多数排序算法都有两个基本的操作： 比较 和 移动 。

2. 在对一组记录（54，38，96，23，15，72，60，45，83）进行直接插入排序时，当把第7个记录60插入到有序表时，为寻找插入位置至少需比较 3 次。

3. 在插入和选择排序中，若初始数据基本正序，则选用 插入 ；若初始数据基本反序，则选用 选择 。

正序时两种方法移动次数均为0，但比较次数量级不同，插入法：n-1即O(n)，选择法：O(n2)

反序时两种方法比较次数量级相同，均为O(n2)，但移动次数不同，插入法：O(n2)，选择法：3(n-1)即O(n)

4. 在堆排序和快速排序中，若初始记录接近正序或反序，则选用 堆排序 ；若初始记录基本无序，则最好选用 快速排序 。

5. 对于n个记录的集合进行冒泡排序，在最坏的情况下所需要的时间复杂度是 O(n2) 。若对其进行快速排序，在最坏的情况下所需要的时间复杂度是 O(n2) 。

6. 对于n个记录的集合进行归并排序，所需要的平均时间是 O(nlog2n)

，所需要的附加空间是 O(n) 。

7． 对于n个记录的表进行2路归并排序，整个归并排序需进行 ┌log2n┐ 趟（遍）。

8. 设要将序列（Q, H, C, Y, P, A, M, S, R, D, F, X）中的关键码按字母序的升序重新排列，则：

冒泡排序一趟扫描的结果是 H C Q P A M S R D F X Y ；

二路归并排序一趟扫描的结果是 H Q C Y A P M S D R F X；

快速排序一趟扫描的结果是 F H C D P A M Q R S Y X ；

堆排序初始建堆的结果是 Y S X R P C M H Q D F A 。（大根堆）

9. 在堆排序、快速排序和归并排序中，

若只从存储空间考虑，则应首先选取 堆排序 方法，其次选取 快速排序方法，最后选取归并排序方法；

若只从排序结果的稳定性考虑，则应 选取归并排序方法；

若只从平均情况下最快考虑，则应选取快速排序方法；

若只从最坏情况下最快并且要节省内存考虑，则应选取堆排序方法。