第一章 绪论

1．下面是几种数据的逻辑结构S=(D，R)，分别画出对应的数据逻辑结构，并指出它们分别属于何种结构。

D={a，b，c，d，e，f} R={r}

(a) r={<a，b>，<b，c>，<c，d>，<d，e>，<e，f>}

1. r={<a，b>，<b，c>，<b，d>，<d，e>，<d，f>}
2. r={<a，b>，<b，c>，<d，a>，<d，b>，<d，e>}

2．分析下列程序段的时间复杂度

(a) for(i=0；i<m；i++)

for(j=0；j<n；j++)

b[i][j]=0；

1. s=0

for(i=0；i<n；i++)

for(j=0；j<n；j++)

s+=b[i][j]；

1. i=1

While(i<n)

i\*=2；

3．在数据结构中，与所使用的计算机无关的是 。

A．存储结构 B．物理结构 C．物理和存储结构 D．逻辑结构

4．非线性结构中每个结点 。

A．无直接前驱结点 B．只有一个直接前驱和直接后继结点

C．无直接后继结点 D．可能有多个直接前驱和多个直接后继结点

5．可以把数据的逻辑结构划分成 。

A．内部结构和外部结构 B．动态结构和静态结构

C．紧凑结构和非紧凑结构 D．线性结构和非线性结构

第二章 线性表

**一、单项选择题**

1．下面关于线性表叙述中，错误的 是\_(1)\_。

(1)：A．顺序表必须占用一片地址连续的存储单元

B．链表不必占用一片地址连续的存储单元

C．顺序表可以随机存取任一元素

D．链表可以随机存取任一元素

2．在表长为n的单链表中，算法时间复杂度为O(n)的操作是 (2)。

(2)：A．查找单链表中第i个结点 B．在p结点之后插入一个结点

C．删除表中第一个结点 D．删除p结点的直接后继结点

3．单链表的存储密度 (3) 。

(3)：A．大于1 B．等于1 C．小于1 D．不能确定

4．在表长为n的顺序表中，算法的时间复杂度为O(1)的操作是 (4) 。

(4)：A．在第n个结点之后插入一个结点 B．在第i个结点前插入一个新结点

C．删除第i个结点 D．求表长

5．在下列链表中不能从当前结点出发访问到其余各结点的是（5）。

(5)：A．单链表 B．单循环链表 C．双向链表 D．双向循环链表

6．在设头、尾指针的单链表中，与长度n有关的操作是（6）。

(6)：A．删除第一个结点 B．删除最后一个结点

C.在第一个结点之前插入一个结点 D.在表尾结点后插入一个结点

7.设p结点是带表头结点的双循环链表中的结点，则

(1)在p结点后插入s结点的语句序列中正确的是 （7）。

（7）：A.s->next=p->next；p->next->prior=s；

p->next=s；s->next=p->next；

B.p—>next=s；S—>next=p—>next；

p—>next—>prior=s；s—>next=p；

C.p->next=s；p—>next—>prior=s；

s->next=p—>next；S—>next=p；

D.p->next->prior=s；p->next=s；

s->next=p->next；s->next=p；

(2)在p结点之前插入s结点的语句序列中正确的是(8)。

(8)：A.s->prior=p->prior；p->prior->next p->prior=s；s->next=p；

B.p->prior=s；p->prior->next=s;

s->prior=p->prior；s->next=p;

C.p->prior->next=s；p->prior=s;

s->prior=p->prior；s->next=p;

D.p->prior=s；s->next=p；

p->prior->next=s；s->prior=p->prior;

8．下列关于链表说法错误的是 (9) 。

(9)：A.静态链表中存取每一个元素的时间相同

B．动态链表中存取每一个元素的时间不同

C．静态链表上插入或删除一个元素不必移动其他元素

D．动态链表上插入或删除一个元素不必移动其他元素

9．在链表中最常用的操作是在最后一个数据元素之后插入一个数据元素和删除第一个数据元素，则最节省运算时间的存储方式是 (10) 。

(10)：A.仅有头指针的单链表 B．仅有头指针的单循环链表

C．仅有头指针的双向链表 D．仅有尾指针的单循环链表

**二、填空题**

1．单链表中每个结点需要两个域，一个是数据域，另一个是 (1) 。

2．链表相对于顺序表的优点是 (2) 和 (3) 操作方便。

3．表长为n的顺序表中，若在第j个数据元素(1≤i≤n+1)之前插入一个数据元素，需要向后移动 (4) 个数据元素；删除第j个数据元素需要向前移动 (5) 个数据元素；在等概率的情况下，插入一个数据元素平均需要移动 (6) 个数据元素，删除一个数据元素平均需要移动 (7) 个数据元素。

4．单链表h为空表的条件是 (8) 。

5．带表头结点的单链表h为空表的条件是 (9) 。

6．在非空的单循环链表h中，某个p结点为尾结点的条件是 (10)。

7．在非空的双循环链表中，已知p结点是表中任一结点，则

(1)在p结点后插入s结点的语句序列是：

s->next=p->next；s->prior=p； (11) ； (12)

(2)在p结点前插入s结点的语句序列是：

s->prior=p->prior；s->next=p； (13) ； (14)

(3)删除p结点的直接后继结点的语句序列是：

q=p->next；p->next=q->next； (15) ；free(q)；

(4)删除p结点的直接前驱结点的语句序列是：

q=p->prior；p->prior=q->prior； (16) ；free(q)；

(5)删除p结点的语句序列是：

p->prior->next=p->next； (17) ；free(q)；

8．在带有尾指针r的单循环链表中，在尾结点后插入p结点的语句序列是 (18) ； (19)；删除第一个结点的语句序列是q=r->next； (20) ；free(q)。

**三、应用题**

1．简述顺序表和链表各自的优点。

2．简述头指针和头结点的作用。 ．

**四、算法设计题**

1．请编写一个算法，实现将x插入到已按数据域值从小到大排列的有序表中。

2．设计一个算法，计算单链表中数据域值为x的结点个数。

3．设计一个用前插法建立带表头结点的单链表的算法。

4．请编写一个建立单循环链表的算法。

5．设计一个算法，实现将带头结点的单链表进行逆置。

6．编写一个算法，实现以较高的效率从有序顺序表A中删除其值在x和y之间(x≤A[i] ≤y)的所有元素。

第三章 栈和队列

**一、选择题**

1．插入和删除只能在表的一端进行的线性表，称为 (1) 。

(1)：A．队列 B．循环队列 C．栈 D．双栈

2．队列操作应遵循的原则是 （2） 。

(2)：A.先进先出 B．后进先出 C．先进后出 D．随意进出

3．栈操作应遵循的原则是 （3） 。

(3)：A．先进先出 B．后进后出 C．后进先出 D．随意进出

4．设队长为n的队列用单循环链表表示且仅设头指针，则进队操作的时间复杂度为 （4） 。

(4)：A．O(1) B．O(log2n) C．O(n) D．O(n2)

5．设栈s和队列q均为空，先将a，b，c，d依次进队列q，再将队列q中顺次出队的元素进栈s，直至队空。再将栈s中的元素逐个出栈，并将出栈元素顺次进队列q，则队列q的状态是 (5) 。

(5)：A．abcd B．dcba C．bcad D．dbca

6．若用一个大小为6的数组来实现循环队列，且当前front和rear的值分别为3和0，当从队列中删除一个元素，再加入两个元素后，front和rear的值分别为 (6) 。

(6)：A．5和1 B．4和2 C．2和4 D．1和5

7．一个栈的入栈序列是a,b,c,d,e,则栈的不可能的输出序列是 （7） 。

(7)：A.edcba B.decba C.dceab D.abcde

**二、填空题**

1．在栈结构中，允许插入、删除的一端称为 (1) ，另一端称为 (2) 。

2．在队列结构中，允许插入的一端称为 (3) ，允许删除的一端称为 (4) 。

3．设长度为n的链队列用单循环链表表示，若只设头指针，则进队和出队操作的时间复杂度分别是 (5) 和 (6) ；若只设尾指针，则进队和出队操作的时间复杂度分别为 (7) 和 (8) 。

4．设用少用一个元素空间的数组A[m]存放循环队列，front指向实际队首，rear指向新元素应存放的位置，则判断队空的条件是 (9) ，判断队满的条件是 (10) ，当队未满时，循环队列的长度是 (11) 。

5.两个栈共享一个向量空间时，可将两个栈底分别设在 (12) 。

**三、应用题**  ．

1．简述线性表、栈和队列有什么异同?

2．循环队列的优点是什么?设用数组A[m]来存放循环队列，如何判断队满和队空。

3．若进栈序列为abcd，请给出全部可能的出栈序列和不可能的出栈序列。

4．设栈s和队列q初始状态为空，元素a，b，c，d，e和f，依次通过栈s，一个元素出栈后即进人队列，若6个元素出队的序列是bdcfea，则栈s的容量至少应该存多少个元素?

5．已知一个中缀算术表达式为 3 + 4 / (25- (6+15))\*8 写出对应的后缀算术表达式（逆波兰表达式）。

**四、算法设计题**

1.已知q是一个非空顺序队列，s是一个顺序栈，请设计一个算法，实现将队列q中所有元素逆置。

2．已知递归函数： 1 当n=0时

# F(n)=

n·F(n/2) 当n>0时

(1)写出求F(n)递归算法；

(2)写出求F(n)的非递归算法。

3．假设以带头结点的循环链表表示队列，并且仅设一个指针指向队尾元素结点（注意不设头指针），试编写相应的队列初始化、入队列和出队列的算法。

第四章 串、数组和广义表

**一、选择题**

1．串的模式匹配是指 (1) 。

(1)：A.判断两个串是否相等

B．对两个串进行大小比较

C．找某字符在主串中第一次出现的位置

D．找某子串在主串中第一次出现的第一个字符位置

2．设二维数组A[m][n]，每个数组元素占用d个存储单元，第一个数组元素的存储地址是如Loc(a[0][0])，求按行优先顺序存放的数组元素a[j][j](0≤i≤m-1，0≤j≤n-1)的存储地址 (2) 。

(2)：A．Loc(a[0][0]+[(i-1)\*n+j-1]\*d

B．Loc(a[0][0])+[i\*n+j]\*d

C．Loc(a[0][0]+[j\*m+i]\*d

D．Loc(a[0][0])+[(j-1)\*m+i-1]\*d

3．设二维数组A[m][n]，每个数组元素占d个存储单元，第1个数组元素的存储地址是Loc(a[0][0])，求按列优先顺序存放的数组元素a[j][j](0≤i≤m-1，0≤j≤n-1)的存储地址 (3) 。

(3)：A．Loc(a[0][0]+[(i-1)\*n+j-1]\*d

B．Loc(a[0][0])+[i\*n+j]\*d

C．Loc(a[0][0]+[(j-1)\*m+i]\*d

D．Loc(a[0][0])+[j\*m+i]\*d

4．已知二维数组A[6][10]，每个数组元素占4个存储单元，若按行优先顺序存放数组元素a[3][5]的存储地址是1000，则a[0][0]的存储地址是 (4) 。

(4)：A．872 B．860 C．868 D．864

5．若将n阶上三角矩阵A，按列优先顺序压缩存放在一维数组F[n(n+1)／2]中，第1个非零元素a11存于F[0]中，则应存放到F[K]中的非零元素aij(1≤i≤n，1≤j≤i的下标i,j与K的对应关系是 (5) 。

(5)：A．i(i+1)/2+j B．i(i-1)/2+j-1

C．j(j+1)／2+j D．j(j-1)／2+i—1

6．若将n阶下三角矩阵A，按列优先顺序压缩存放在一维数组F[n(n+1)／2]中，第一个非零元素a11，存于F[0]中，则应存放到F[K]中的非零元素aij(1≤j≤n，1≤j≤i)的下标i，i与K的对应关系是 (6) 。

(6):A.(2n-j+1)j/2+I-j B.(2n-j+2)(j-1)/2+i-j

C．(2n-i+1)i/2+j-I D．(2n-i+2)i／2+j-i

7．设有10阶矩阵A，其对角线以上的元素aij(1≤j≤10，1<i<j)均取值为-3，其他矩阵元素为正整数，现将矩阵A压缩存放在一维数组F[m]中，则m为 (7) 。

(7)：A．45 B．46 C．55 D.56

8．设广义表L=(a，b，L)其深度是 (8) 。

(8)：A．3 B．∞ C．2 D．都不对

9．广义表B：(d)，则其表尾是 (9) ，表头是 (10) 。

(9)—(10)：A．d B．() C．(d) D．(())

10．下列广义表是线性表的有 （11） 。

(11)：A．Ls=(a，(b，c)) B．Ls=(a，Ls)

C．Ls=(a，b) D．Ls=(a，(()))

11.一个非空广义表的表尾 (12) 。

(12)：A．只能是子表 B．不能是子表

C．只能是原子元素 D．可以是原子元素或子表

12．已知广义表A=((a，(b，c))，(a，(b，c)，d))，则运算head

(head(tail(A)))的结果是 (13) 。

(13)：A．a B．(b，c) C．(a，(b，c)) D．d

**二、填空题**

1．两个串相等的充分必要条件是 (1) 。

2．空串是 (2) ，其长度等于 (3) 。

3．设有串S=”good”，T=”morning”，求：

(1)concat(S，T)= (4) ；

(2)substr(T，4，3)= (5) ；

(3)index(T，”n”)= (6) ；

(4)replace(S，3，2，”to”)= (7) 。

4．若n为主串长，m为子串长，则用简单模式匹配算法最好情况下，需要比较字符总数是 (8) ，最坏情况下，需要比较字符总数是 (9) 。

5．设二维数组A[8][10]中，每个数组元素占4个存储单元，数组元素a[2][2]按行优先顺序存放的存储地址是1000，则数组元素a[0][0]的存储地址是 (10) 。

6．设有矩阵

1. -3 -3 -3

4 2 -3 -3

6 5 7 -3

1 3 9 11

A=

压缩存储到一维数组F[m]中，则m为 (11) ，-3应存放到F[k1]中，k1为(12)，元素aij(1≤i≤4，1≤j≤i)按行优先顺序存放到F[k2]中，k2为 (13) ，按列优先顺序存放到F[k3]中，k3为 (14) 。

7．广义表Ls=(a，(b)，((c，(d))))的长度是 (15) ，深度是 (16) ，表头是 (17) ，表尾是 (18) 。

8．稀疏矩阵的压缩存储方法通常有两种，分别是 (19) 和 (20) 。

9．任意一个非空广义表的表头可以是原子元素，也可以是 (21) ，而表尾必定是 (22) 。

**三、应用题**

1．已知S=”(xyz)+\*”试利用联接(concat(S，T))，取子串(substr(S，i，j))和置换(replace(S，i，j，T))基本操作将S转化为T=”(x+2)\*y”。

2．设串S的长度为n，其中的字符各不相同，求S中互异的非平凡子串(非空且不同于S本身)的个数。

3．设模式串T=”abcaaccbaca”，请给出它的next函数及next函数的修正值nextval之值。

4．特殊矩阵和稀疏矩阵哪一种压缩存储会失去随机存储功能?

5．设n阶对称矩阵A压缩存储于一维数组F[m]中，矩阵元素aij(1≤i≤n，1≤j≤n)，存于F[k](0≤k<m)中，当用行优先顺序转换时的对应关系是什么?用列优先顺序转换时的对应关系是什么?

第五章 树和二叉树

**一、单项选择题**

1．关于二叉树的下列说法正确的是 (1)。

(1)：A．二叉树的度为2 B．二叉树的度可以小于2

C．每一个结点的度都为2 D．至少有一个结点的度为

2．设深度为h(h>0)的二叉树中只有度为0和度为2的结点，则此二叉树中所含的结点总数至少为 （2）。

(2)A．2h B．2h-1 C．2h+1 D．h+1

3．在树中，若结点A有4个兄弟，而且B是A的双亲，则B的度为 (3) 。

(3)：A．3 B．4 C．5 D．6

4．若一棵完全二叉树中某结点无左孩子，则该结点一定是 (4) 。

(4)：A.度为1的结点 B．度为2的结点 C．分支结点 D．叶子结点

5．深度为k的完全二叉树至多有 (5) 个结点，至少有 (6) 个结点。

(5)-(6)：A．2k-1-1 B．2k-1  C．2k-1 D．2k

6．前序序列为ABC的不同二叉树有 (7) 种不同形态。

(7)：A．3 B．4 C．5 D．6

7．若二叉树的前序序列为DABCEFG，中序序列为BACDFGE，则其后序序列为 (8) ，层次序列为 (9) 。

(8)-(9)：A．BCAGFED B．DAEBCFG C．ABCDEFG D．BCAEFGD

8．在具有200个结点的完全二叉树中，设根结点的层次编号为1，则层次编号为60的结点，其左孩子结点的层次编号为 (10) ，右孩子结点的层次编号为 (11) ，双亲结点的层次编号为 (12)。

(10)-(12)：A．30 B．60 C．120 D．121

9．遍历一棵具有n个结点的二叉树，在前序序列、中序序列和后序序列中所有叶子结点的相对次序 (13) 。

(13)：A.都不相同 B．完全相同 C．前序和中序相同 D．中序与后序相同

10．在由4棵树组成的森林中，第一、第二、第三和第四棵树组成的结点个数分别为30，10，20，5，当把森林转换成二叉树后，对应的二叉树中根结点的左子树中结点个数为 （14） ，根结点的右子树中结点个数为 (15) 。

(14)—(15)：A．20 B．29 C．30 D．35

11.具有n个结点(n>1)的二叉树的前序序列和后序序列正好相反，则该二叉树中除叶子结点外每个结点 (16) 。

(16)：A.仅有左孩子 B．仅有右孩子 C．仅有一个孩子 D．都有左、右孩子

12．判断线索二叉树中p结点有右孩子的条件是 (17) 。

(17)：A.p!=NULL B．p->rchild!=NULL C．p->rtag=0 D．p->rtag=1

13．将一棵树转换成二叉树，树的前根序列与其对应的二叉树的 (18) 相等。树的后根序列与其对应的二叉树的 (19)相同。

(18)—(19)：A．前序序列 B．中序序列 C．后序序列 D．层次序列

14．设数据结构(D，R)，D={dl，d2，d3，d4，d5，d6}，R={<d4，d2>，<d2，d1>，<d2，d3>，<d4，d6>，<d6，d5>}，这个结构的图形是 (20) ；用 (21) 遍历方法可以得到序列{d1，d2，d3，d4，d5，d6}。

(20)：A．线性表 B．二叉树C．队列 D．栈

(21)：人前序 B．中序 C．后序 D．层次

15．对于树中任一结点x，在前根序列中序号为pre(x)，在后根序列中序号为post(x)，若树中结点x是结点y的祖先，下列 （22） 条件是正确的。

(22)：A.pre(x)<pre(y)且post(x)<post(y)

B．pre(x)<pre(y)且post(x)>post(y)

C. pre(x)>pre(y)且post(x)<post(y)

D．pre(x)>pre(y)且post(x)>post(y)

16．每棵树都能惟一地转换成对应的二叉树，由树转换的二叉树中，一个结点N的左孩子是它在原树对应结点的 (23) ，而结点N的右孩子是它在原树里对应结点的 (24) 。

(23)—(24)：A．最左孩子 B．最右孩子 C．右邻兄弟 D．左邻兄弟

17．二叉树在线索化后，仍不能有效求解的问题是 (25) 。

(25)：A．前序线索树中求前序直接后继结点

B．中序线索树中求中序直接前驱结点

C．中序线索树中求中序直接后继结点

D．后序线索树中求后序直接后继结点

18．一棵具有124个叶子结点的完全二叉树，最多有 (26)个结点。

(26)：A．247 B．248 C．249 D．250 。

19．实现任意二叉树的后序遍历的非递归算法而不使用栈结构，最有效的存储结构是采用 （27） 。 (27)：A. 二叉链表 B．孩子链表 C．三叉链表 D．顺序表

**二、填空题**

1.树中任意结点允许有 （1）孩子结点，除根结点外，其余结点 （2） 双亲结点。

2.若一棵树的广义表表示为A(B(E，F)，C(C(H，I，J，K)，L)，D(M(N)))。则该树的度为 (3) ，树的深度为 (4) ，树中叶子结点个数为 (5) 。

3．若树T中度为1、2、3、4的结点个数分别为4、3、2、2，则T中叶子结点的个数是 (6) 。

4．一棵具有n个结点的二叉树，若它有m个叶子结点，则该二叉树中度为1的结点个数是 (7) 。

5．深度为k(k>0)的二叉树至多有 (8) 个结点，第i层上至多有 (9) 个结点。

6．已知二叉树有52个叶子结点，度为1的结点个数为30则总结点个数为 (10) 。

7．已知二叉树中有30个叶子结点，则二叉树的总结点个数至少是 （11） 。

8．高度为6的完全二叉树至少有 (12) 个结点。

9．一个含有68个结点的完全二叉树，它的高度是 (13) 。

10．已知一棵完全二叉树的第6层上有6个结点(根结点的层数为1)，则总的结点个数至少是 (14) ，其中叶子结点个数是 （15） 。

11．已知完全二叉树第6层上有10个叶子结点，则这棵二叉树的结点总数最多是 （16） 。

12．一棵树转换成二叉树后，这棵二叉树的根结点一定没有 (17) 孩子，若树中有m个分支结点，则与其对应的二叉树中无右孩子的结点个数为 （18） 。

13．若用二叉链表示具有n个结点的二叉树，则有 （19） 个空链域。

14．具有m个叶子结点的哈夫曼树，共有 （20）个结点。

15．树的后根遍历序列与其对应的二叉树的 （21） 遍历序列相同。

16．线索二叉树的左线索指向其 （22） ，右线索指向其 （23） 。

**三、应用题**

1．具有n个结点的满二叉树的叶子结点个数是多少?

2．列出前序遍历序列是ABC的所有不同的二叉树。

3．已知二叉树的层次遍历序列为ABCDEFGHIJK，中序序列为DBGEHJACIKF，请构造一棵二叉树。

4．已知二叉树的中序遍历序列是ACBDGHFE，后序遍历序列是ABDCFHEG，请构造一棵二叉树。

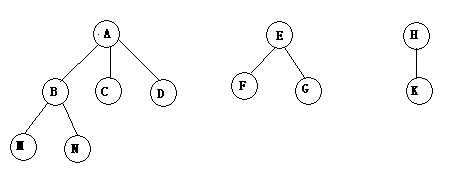
5．已知二叉树的前序、中序和后序遍历序列如下，其中有一些看不清的字母用\*表示，请先填写\*处的字母，再构造一棵符合条件的二叉树，最后画出带头结点的中序线索链表。

(1)前序遍历序列是：\*BC\*\*\*G\*

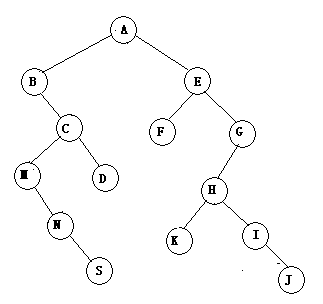
(2)中序遍历序列是：CB\*EAGH\*

(3)后序遍历序列是：\*EDB\*\*FA

6．将下图所示的森林转换成一棵二叉树，并画出这棵二叉树的顺序存储结构。



7．将下图所示的二叉树还原成森林。



8．对于给定的一组权值{3，5，6，7，9}，请构造相应的哈夫曼树，并计算其带权路径长度。

**四、算法设计题**

1．请设计一个算法，求以孩子兄弟链表表示的树中叶子结点个数。

2．请编写一个算法，实现将以二叉链表存储的二叉树中所有结点的左、右孩子进行交换。

3．请编写一个算法，将以二叉链表存储的二叉树输出其广义表表示形式。

4．请编写一个算法，判断以二叉链表存储的二叉树是否为完全二叉树。

5．假设二叉树采用链接存储方式存储，编写一个二叉树先序遍历和后序遍历的非递归算法。..

6．已知一棵二叉树用二叉链表存储，t指向根结点，p指向树中任一结点。请编写一个非递归算法，实现求从根结点t到结点p之间的路径。

第六章 图

**一、单项选择题**

1．下面关于图的存储结构的叙述中正确的是 (1) 。

(1)：A．用邻接矩阵存储图占用空间大小只与图中顶点有关，与边数无关

B．用邻接矩阵存储图占用空间大小只与图中边数有关，而与顶点数无关

C．用邻接表存储图占用空间大小只与图中顶点数有关，而与边数无关

D．用邻接表存储图占用空大小只与图中边数有关，而与顶点数无关

2．下面关于对图的操作的说法不正确的是 (2) 。

(2)：A:寻找关键路径是关于带权有向图的操作

B．寻找关键路径是关于带权无向图的操作

C．连通图的生成树不一定是惟一的

D．带权无向图的最小生成树不一定是惟一的

3．下面的各种图中，哪个图的邻接矩阵一定是对称的 (3) 。

(3)：A．AOE网 B．AOV网 C．无向图 D．有向图

4．在AOE网中关于关键路径叙述正确的是 (4) 。

(4)：A．从开始顶点到完成顶点的具有最大长度的路径，关键路径长度是完成整个工程所需的最短时间

B．从开始顶点到完成顶点的具有最小长度的路径，关键路径长度是完成整个工程所需的最短时间

C．从开始顶点到完成顶点的具有最大长度的路径，关键路径长度是完成整个工程所需的最长时间

D．从开始顶点到完成顶点的具有最小长度的路径，关键路径长度是完成整个工程所需的最短时间

5．一个具有n个顶点e条边的图中，所有顶点的度数之和等于 （5）。

(5)：A．n B．2n C．e D．2e

6．具有8个顶点的无向图最多有 (6) 条边。

(6)：A．8 B．28 C．56 D．72

7．具有8个顶点的有向图最多有 (7) 条边。

(7)：A．8 B．28 C．56 D．78

8．深度优先遍历类似于二叉树的 (8) 。

(9)：A．前序遍历 B．中序遍历 C．后序遍历 D．层次遍历

9．广度优先遍历类似于二叉树的 (9) 。

(9)：A．前序遍历 B．中序遍历 C．后序遍历 D．层次遍历

10．任一个连通图的生成树 (10) 。

（l0）：A．可能不存在 B．只有一棵 C．一棵或多棵 D．一定有多棵

11．下列关于连通图的BFS和DFS生成树高度论述正确的是 （11）。

(11)：A．BFS生成树高度<DFS生成树的高度

B．BFS生成树高度≤DFS生成树的高度

C．BFS生成树高度>DFS生成树的高度

D．BFS生成树高度≥DFS生成树的高度

12．G是一个非连通无向图，共有28条，则该图至少有解 (12) 个顶点。

(12)：A．7 B．8 C．9 D.10

**二、判断题**

1．一个图的邻接矩阵表示是惟一的。 ( )

2．一个图的邻接表表示是惟一的。 ( )

3．无向图的邻接矩阵一定是对称矩阵。 ( )

4．有向图的邻接矩阵一定是对称矩阵。 ( )

5．有向图用邻接表表示，顶点vi的出度是对应顶点vj链表中结点个数。 ( )

6．有向图用邻接表表示，顶点vi的度是对应顶点vj链表中结点个数。 ( )

7．有向图用邻接矩阵表示，删除所有从顶点i出发的弧的方法是，将邻接矩阵的i行全部元素置为0。 ( )

8．若从无向图中任一顶点出发，进行一次深度优先搜索，就可以访问图中所有顶点，则该图一定是连通的。 ( )

9．在非连通图的遍历过程中，调用深度优先搜索算法的次数等于图中连通分量的个数。 ( )

10．具有n个顶点的有向强连通图的邻接矩阵中至少有n个小于∞的非零元素。 ( )

11．一个连通图的生成树是一个极小连通子图。 ( )

12．在有数值相同的权值存在时，带权连通图的最小生成树可能不惟一。 ( )

13．在AOE网中仅存在一条关键路径。 ( )

14．若在有向图的邻接矩阵中，主对角线以下的元素均为0，则该图一定存在拓扑有序序列。 ( )

15．若图C的邻接表表示时，表中有奇数个边结点，则该图一定是有向图。 ( )

16．在AOE网中，任何一个关键活动提前完成，整个工程都会提前完成。 ( )

17．图的深度优先遍历序列一定是惟一的。 ( )

18．有向无环图的拓扑有序序列一定是惟一的。 ( )

19．拓扑排序输出的顶点个数小于图中的顶点个数，则该图一定存在环。 ( )

**三、填空题**

1．在一个具有n个顶点的完全无向图和完全有向图中分别包含有 (1) 和 (2) 条边。

2．具有n个顶点的连通图至少具有 (3) 条边。

3．具有n个顶点e条边的有向图和无向图用邻接表表示，则邻接表的边结点个数分别为（4）和 （5） 条。

4．在有向图的邻接表和逆邻接表中，每个顶点链表中链接着该顶点的所有 （6） 和 （7）结点。

5．若n个顶点的连通图是一个环，则它有 (8) 棵生成树。

6．图的逆邻接表存储结构只适用于 (9) 。

7．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是 （10） ，广度优先算法的时间复杂度是 （11） 。

8．n个顶点e条边的图，采用邻接表存储，广度优先遍历算法的时间复杂度是 （12） ，深度优先遍历算法的时间复杂度是 （13） 。

9．若要求一个稀疏图的最小生成树，最好用 (14) 算法求解。

10．若要求一个稠密图的最小生成树，最好用 （15） 算法求解。

**四、应用题**

1．已知图C=(V，E)，其中V={a，b，c，d，e}，E={(a，b)，{a，c}，{a，d}，(b，c)，(d，c)，(b，e)，(c，e)，（d，e）}要求：

(1)画出图G；

(2)给出图G的邻接矩阵；

(3)给出图G的邻接表；

(4)给出图G的所有拓扑有序序列。

2．已知带权无向图的邻接表见下图，要求：

|  |  |
| --- | --- |
| a |  |
| b |  |
| c |  |
| d |  |
| e |  |
| f |  |
| g |  |

0

1

2

3

4

5

6

|  |  |  |
| --- | --- | --- |
| 1 | 9 |  |

|  |  |  |
| --- | --- | --- |
| 3 | 3 |  |

|  |  |  |
| --- | --- | --- |
| 4 | 2 |  |

|  |  |  |
| --- | --- | --- |
| 5 | 8 | ^ |

|  |  |  |
| --- | --- | --- |
| 0 | 9 |  |

|  |  |  |
| --- | --- | --- |
| 2 | 2 |  |

|  |  |  |
| --- | --- | --- |
| 5 | 3 | ^ |

|  |  |  |
| --- | --- | --- |
| 1 | 2 |  |

|  |  |  |
| --- | --- | --- |
| 3 | 7 |  |

|  |  |  |
| --- | --- | --- |
| 6 | 8 | ^ |

|  |  |  |
| --- | --- | --- |
| 2 | 7 |  |

|  |  |  |
| --- | --- | --- |
| 4 | 4 |  |

|  |  |  |
| --- | --- | --- |
| 6 | 9 | ^ |

|  |  |  |
| --- | --- | --- |
| 0 | 2 |  |

|  |  |  |
| --- | --- | --- |
| 3 | 4 |  |

|  |  |  |
| --- | --- | --- |
| 5 | 4 |  |

|  |  |  |
| --- | --- | --- |
| 6 | 1 | ^ |

|  |  |  |
| --- | --- | --- |
| 0 | 8 |  |

|  |  |  |
| --- | --- | --- |
| 1 | 3 |  |

|  |  |  |
| --- | --- | --- |
| 4 | 4 |  |

|  |  |  |
| --- | --- | --- |
| 6 | 5 | ^ |

|  |  |  |
| --- | --- | --- |
| 2 | 8 |  |

|  |  |  |
| --- | --- | --- |
| 3 | 9 |  |

|  |  |  |
| --- | --- | --- |
| 4 | 1 |  |

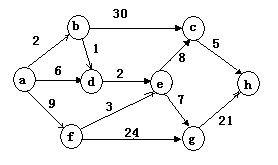
|  |  |  |
| --- | --- | --- |
| 5 | 5 | ^ |

|  |  |  |
| --- | --- | --- |
| 0 | 3 |  |

(1)画出图G。

(2)各画一棵从顶点a出发的深度优先生成树和广度优先生成树。

(3)给出用prim算法从顶点a出发构造最少生成树的过程。

 (4)给出用Kruscal算法构造最小生成树的过程。

3．已知带权有向图如右图所示，要求：

(1)给出图G的邻接矩阵；

(2)给出图G的一个拓扑有序序列；

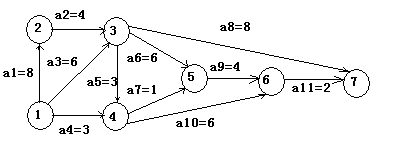
(3)求从顶点a出发到其余各顶点的最短路径。

4．已知带权有向图G见下图，图中边上的权值为完成活动ai需要的天数，要求：

(1)求每项活动的最早和最晚开工时间；

(2)完成此项工程最少需要多少天；

(3)给出图G的关键路径。



**四、算法题**

1.编写根据无向图G的邻接表，判断图G是否连通的算法。

2．编写根据有向图的邻接表，分别设计实现以下要求的算法：

(1)求出图G中每个顶点的出度；

(2)求出图G中出度最大的一个顶点，输出该顶点编号；

(3)计算图G中出度为0的顶点数；

(4)判断图G中是否存在边〈i，j〉。

第七章 查找

**一、单项选择题**

1．在长度为n的线性表中进行顺序查找，在等概率的情况下，查找成功的平均查找长度是 (1) 。

(1)：A．n B．n(n+1)／2 C．(n-1)／2 D．(n+1)／2

2．在长度为n的顺序表中进行顺序查找，查找失败时需与键值比较次数是 (2) 。

(2)：A．n B．1 C．n-1 D．n+l

3．对线性表进行顺序查找时，要求线性表的存储结构是 （3） 。

(3)：A．倒排表 B．索引表 C．顺序表或链表 D．散列表

4．对线性表用二分法查找时要求线性表必须是 （4） 。

(4)：A．顺序表 B．单链表 C．顺序存储的有序表 D．散列表

5．在有序表A[80]上进行二分法查找，查找失败时，需对键值进行最多比较次数是 （5） 。

(5)：A．20 B．40 C．10 D．7

6．在长度为n的有序顺序表中，采用二分法查找，在等概率的情况下，查找成功的平均查找长度是 （6）。

(6)：A. O(n2) B．O(nlog2n) C．O(n) D．O(log2n)

7．设顺序表为{4，6，12，32，40，42，50，60，72，78，80，90，98}，用二分法查找72，需要进行的比较次数是 (7) 。

(7)：A．2 B．3 C．4 D．5

8．如果要求一个线性表既能较快地查找，又能适应动态变化的要求，则应采用的查找方法是 (8) 。

(8)：A．顺序查找 B．二分法查找 C．分块查找 D．都不行

9．在采用线性探查法处理冲突的散列表中进行查找，查找成功时所探测位置上的键值 (9) 。

(9)：A．一定都是同义词 B．一定都不是同义词

C．不一定是同义词 D．无任何关系

10．在查找过程中，若同时还要做插入、删除操作，这种查找称为 (10) 。

(10)：A．静态查找 B．动态查找 C．内部查找D． 外部查找

**二、填空题**

1．在有序表A[30]上进行二分查找，则需要对键值进行4次、5次比较查找成功的记录个数分别为 (1) 、 (2) ，在等概率的情况下，查找成功的平均查找长度是 (3) 。

2．散列法存储的基本思想是由 (4) ，确定记录的存储地址。

3．对一棵二叉排序树进行 (5) 遍历，可以得到一个键值从小到大次序排列的有序序列。

4．对有序表A[80]进行二分查找，则对应的判定树高度为 (6) ．，判定树前6层的结点个数为 (7) ，最高一层的结点个数是 (8) ，查找给定值K，最多与键值比较 (9) 次。 5．对于表长为n的线性表要进行顺序查找，则平均时间复杂度为 (10) ，若采用二分查找，则平均时间复杂度为 (11) 。

6．在散列表中，装填因子。值越大，则插入记录时发生冲突的可能性 (12) 。

7．在散列表的查找过程中与给定值K进行比较的次数取决于 (13) 、 (14) 和 (15)。

8．在使用分块查找时，除表本身外，尚需建立一个 (16) ，用来存放每一块中最高键值及 (17) 。

9．若表中有10000个记录，采用分块查找时，用顺序查找确定记录所在的块，则分成 (18)块最好，每块的最佳长度 (19) ，在这种情况下，查找成功的平均检索长度为 (20) 。

10．用n个键值构造一棵二叉排序树，最低高度是 (21) 。

11．设有散列函数H和键值k1、k2，若k1≠k2，且H(k1)= H(k2)，则称k1、k2是 (22)。

12．设有n个键值互为同义词，若用线性探查法处理冲突，把这n个键值存于表长为m(m>n)的散列表中，至少要进行 (23) 次探查。

13.高度为6的平衡二叉排序树，其每个分支结点的平衡因子均为0，则该二叉树共有 (24)个结点。

14．m阶B-树，除根结点外的分支结点最多有 (25) 棵子树，最少有 (26) 棵子树，最多有 (27) 个键值，最少有 (28) 个键值。

15．m阶B+树，除根结点外的每个结点最多有 (29) 棵子树值，最少有 (30) 棵子树。

**三、应用题**

1．画出对长度为13的有序表进行二分查找的一棵判定树，并求其等概率时查找成功的平均查找长度。查找失败时需对键值的最多比较次数。

2．将整数序列{8，6，3，1，2，5，9，7，4}中的数依次插入到一棵空的二叉排序树中，求在等概率情况下，查找成功的平均查找长度，查找失败时对键值的最多比较次数。再画出从二叉排序树中删除结点6和8的结果。

3．将整数序列{8，6，3，1，2，5，9，7，41中的数依次插入到一棵空的平衡二叉排序树中，求在等概率情况下，查找成功的平均检索长度，查找失败时对键值的最多比较次数o

4．按不同的输入顺序输入4，5，6，建立相应的不同形态的二叉排序树。

5．设有键值序列{25，40，33，47，12，66，72，87，94，22，5，58}散列表长为12，散列函数为H(key)=key％11，用拉链法处理冲突，请画出散列表，在等概率情况下，求查找成功的平均查找长度和查找失败的平均查找长度。

6．已知一个散列函数为H(key)=key％13，采用双散列法处理冲突，H1(key)=key％11+1，探查序列为di=(H(key)+i\* H1(key))％m，i=1，2，…m-1，散列表见表1，要求回答下列问题：

(1)对表中键值21，57，45和50进行查找时，所需进行的比较次数各为多少?

(2)在等概率情况下查找时，查找成功的平均查找长度是多少?

0 1 2 3 4 5 6 7 8 9 10 11 12

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 50 | 21 |  | 57 |  | 45 | 37 |  |

表1 散列表

**四、算法设计题**

1．设二叉树用二叉链表表示，且每个结点的键值互不相同，请编写判别该二叉树是否为二叉排序树的非递归算法。

第八章 排序

**一、单项选择题**

1．对n个不同的记录按排序码值从小到大次序重新排列，用冒泡(起泡)排序方法，初始序列在 （1） 情况下，与排序码值总比较次数最少，在 （2） 情况下，与排序码值总比较次数最多；用直接插入排序方法，初始序列在 （3） 情况下，与排序码值总比较次数最少，在 （4） 情况下，与排序码值总比较次数最多；用快速排序方法在 （5） 情况下，与排序码值总比较次数最少，在 （5） 情况下与排序码值总比较次数最多。

(1)-(6)：

A．按排序码值从小到大排列 B．按排序码值从大到小排列

C．随机排列(完全无序) D．基本按排序码值升序排列

2．用冒泡排序方法对n个记录按排序码值从小到大排序时，当初始序列是按排序码值从大到小排列时，与码值总比较次数是 （7） 。

(7)：A．n-1 B．n C．n+1 D．n(n-1)／2

3．下列排序方法中，与排序码值总比较次数与待排序记录的初始序列排列状态无关的是 (8) 。

(8)：A．直接插入排序 B．冒泡排序 C．快速排序 D．直接选择排序

4．将6个不同的整数进行排序，至少需要比较 (9) 次，至多需要比较 (10) 次。

(9)-(10)：A．5 B．6 C．15 D．21

5．若需要时间复杂度在O(nlog2n)内，对整数数组进行排序，且要求排序方法是稳定的，则可选择的排序方法是 (11) 。

(11)：A．快速排序 B．归并排序 C．堆排序 D．直接插入排序

6．当待排序的整数是有序序列时，采用 (12) 方法比较好，其时间复杂度为O(n)，而采用 （13）方法却正好相反，达到最坏情况下时间复杂度为O(n2)；无论待排序序列排列是否有序，采用 （14）方法的时间复杂度都是O(n2)。

(12)-(14)：A．快速排序 B．冒泡排序 C．归并排序 D．直接选择排序

7．堆是一种 (15) 排序。

(15)：A．插入 B．选择 C．交换 D．归并

8．若一组记录的排序码值序列为{40，80，50，30，60，70}，利用堆排序方法进行排序，初建的大顶堆是 (16) 。

(16)：A．80，40，50，30，60，70 B．80，70，60，50，40，30

C．80，70，50，40，30，60 D．80，60，70，30，40，50

9．若一组记录的排序码值序列为{50，80，30，40，70，60}利用快速排序方法，以第一个记录为基准，得到一趟快速排序的结果为 （17） 。

(17)：A．30，40，50，60，70，80 B．40，30，50，80，70，60

C．50，30，40，70，60，80 D．40，50，30，70，60，80

10．下列几种排序方法中要求辅助空间最大的是 （18） 。

(18)：A．堆排序 B．直接选择排序 C．归并排序 D．快速排序

11．已知A[m]中每个数组元素距其最终位置不远，采用下列 (19) 排序方法最节省时间。

(19)：A．直接插入 B．堆 C．快速 D．直接选择 12．设有10000个互不相等的无序整数，若仅要求找出其中前10个最大整数，最好采用 （20） 排序方法。

(20)：A．归并 B．堆 C．快速 D．直接选择

13．在下列排序方法中不需要对排序码值进行比较就能进行排序的是 （21） 。

(21)：A：基数排序 B．快速排序 C．直接插入排序 D．堆排序

14．给定排序码值序列为{F，B，J，C，E，A，I，D，C，H}，对其按字母的字典序列的次序进行排列，希尔(Shell)排序的第一趟(d1=5)结果应为 （22） ，冒泡排序(大数下沉)的第一趟排序结果应为 （23） ，快速排序的第一趟排序结果为 （24） ，二路归并排序的第一趟排序结果是 （25） 。

(22)-(25)：A．{B，F，C，J，A，E，D，I，C，H}

B．{C，B，D，A，E，F，I，C，J，H}

C．{B，F，C，E，A，I，D，C，H，J}

D．{A，B，D，C，E，F，I，J，C，H}

**二、填空题**

1．内部排序方法按排序采用的策略可划分为五类： （1）、 （2）、 （3）、 （4）、 （5）。

2．快速排序平均情况下的时间复杂度为 （6） ，其最坏情况下的时间复杂度为 （7）。

3．当待排序的记录个数n很大时，应采用平均时间复杂度为 （8）即 （9）、 （10）、 （11），在这些方法中当排序码值是随机分布时，采用 （12） 排序方法的平均时间复杂度最小。当希望排序方法是稳定时，应采用 (13) 排序方法，若只从节省空间考虑，最节省空间的是 （14） 方法。

4．对一组整数{60，40，90，20，10，70，50，80}进行直接插入排序时，当把第7个整数50插入到有序表中时，为寻找插人位置需比较 （15）次。

5．从未排序序列中挑选最小(最大)元素，并将其依次放到已排序序列的一端，称为 （16）排序。

6．对n个记录进行归并排序，所需要的辅助存储空间是 （17） ，其平均时间复杂度是 （18） ，最坏情况下的时间复杂度是 （19） 。

7．对n个记录进行冒泡排序，最坏情况下的时间复杂度是 （20） 。

8．对20个记录进行归并排序时，共需进行 （21） 趟归并，在第三趟归并时是把最大长度为 （22） 的有序表两两归并为长度为 （23） 的有序表。

**三、应用题**

1．举例说明本章介绍的排序方法中哪些是不稳定的?

2．已知排序码值序列{17，18，60，40，7，32，73，65，85}，请写出冒泡排序每一趟的排序结果。

3．对于排序码值序列{10，18，14，13，16，12，11，9，15，8}，给出希尔排序(dl=5，d2=2，d3=1)的每一趟排序结果。

4．判断下列序列是否为大顶堆?若不是，则把它们调整为大顶堆。

(1){90，86，48，73，35，40，42，58，66，20}

(2){12，70，34，66，24，56，50，90，86，36}

**四、算法设计题**

1．在带表头结点的单链表上，编写一个实现直接选择排序的算法。

2．请编写一个快速排序的非递归算法。

附录：

大连理工大学2002年硕士入学试题

数据结构部分(共50分)

一、算法填空题(20分)

1．对以下函数填空，实现将头指针为h的单链表逆置，即原链表的第一个结点变成逆置后新链表的最后一个结点，原链表的第二个结点变成新链表的倒数第二个结点，如此等等，直到最后一个结点作为新链表的第一个结点，并返回指向该结点的指针。设单链表结点类型的定义为

typedef struct node

{int data；

strcut node \*next；

}NODE；

NODE \*dlbnz(NODE\*h)

{ NODE \*p，\*q；

q=NULL；

while(h)

p=h；

h=h->next；

；

；

}

return q；

}

2．假设算术表达式由字符串b表示，其中可以包含三种括号：圆括号和方括号及花括号，其嵌套的顺序随意，如{[]([])}。请对以下函数填空，实现判别给定表达式中所含括号是否正确配对出现的算法。

#define M 10

int khjc(char \*b)

{ char sM}；

int i，j=0，f=1；

j=0；

for(i=0；f&&b[i]!=’\0’；i++)

{ switch(b[i])

{ case ’(’： ；break；

case ’[’： ；break；

case ‘{’： ；break；

case ’)’：

case ’]’：

case ’[’：if(j==0;||b[I]!= ) f=0；

}

}

return f&& ；

}

3．对以下函数填空，实现以带头结点的单链表h为存储结构的直接选择排序。设单链表结点类型的定义为

typedef struct node

{ int key；

struct node \*next；

}NODE；

void pxx(NODE \*h)

{ NODE \*p，\*q，\*m；

int z；

p=h->next；

while(p!=NULL)

{ q=p->next；

m=p；

while(q!=NULL)

{ if(m->key>q->key) ；

；

}

if(p!=m)

{ x=p->key；

p->key=m->key；

m->key=x；

}

；}

}

二、算法设计题(30分)

请用类C或类PASCAL语言设计实现下列功能的算法。

1．设二叉排序树以二叉链表为存储结构，请编写一个非递归算法，从大到小输出二叉排序树中所有其值不小于X的键值。(10分)

2．设由n个整数组成一个大根堆(即第一个数是堆中的最大值)，请编写一个时间复杂度为O(log2n)的算法，实现将整数X插入到堆中，并保证插入后仍是大根堆。(10分)

3．请编写一个算法，判断含n个顶点和e条边的有向图中是否存在环。并分析算法的时间复杂度。(10分)

大连理工大学2003年硕士入学试题

数据结构部分(共75分)

一、回答下列问题(20分)

1．循环队列用数组A[0．．m—1)存放其数据元素。设tail指向其实际的队尾，front指向其实际队首的前一个位置，则当前队列中的数据元素有多少个?如何进行队空和队满的判断?

2．设散列表的地址空间为0~10，散列函数为H(key)=key％11(％为求余函数)，采用线性探查法解决冲突，并将键值序列{15，36，50，27，19，48}依次存储到散列表中，请画出相应的散列表；当查找键值48时需要比较多少次?

3．什么是m阶B-树?在什么情况下向一棵m阶B-树中插入一个关键字会产生结点分裂?在什么情况下从一棵m阶B-树中删除一个关键字会产生结点合并?

4．什么是线索二叉树?一棵二叉树的中序遍历序列为由djbaechif，前序遍历序列为abdjcefhi，请画出该二叉树的后序线索二叉树。

二、请用类C或类PASCAL语言进行算法设计，并回答相应问题。(45分)

1．设计一非递归算法采用深度优先搜索对无向图进行遍历，并对算法中的无向图的存储结构予以简单说明。

2．用链式存储结构存放一元多项式Pn(x)=P1xel+P2xe2+…+Pnxen，其中Pi是指数为ei的项的非零系数，且满足0≤e1≤e2…<en=n。请设计算法将多项式B=Pn2(x)加到多项式A=Pn1(x)，同时对算法及链表的结点结构给出简单注释。要求利用Pnl(x)和Pn2(x)的结点产生最后求得的和多项式，不允许另建和多项式的结点空间。

3．(1){Rl，R2，…，Rn}为待排序的记录序列，请设计算法对{R1，R2，…，Rn}按关键字的非递减次序进行快速排序。

(2)若待排序的记录的关键字集合是{30，4，48，25，95，13，90，27，18)，请给出采用快速排序的第一趟、第二趟排序结果。

(3)若对(2)中给出的关键字集合采用堆排序，请问初建的小根堆是什么?

(4)当给定的待排序的记录的关键字基本有序时，应采用堆排序还是快速排序?为什么?

三、算法填空(10分)

1．一棵树以孩子兄弟表示法存储，递归算法numberofleaf计算并返回根为r的树中叶子结点的个数(NULL代表空指针)。

typedef struct node{struct node \*firstchild，\*nextbrother；}TFNNode；

int numberofleaf(TFNNode \*r)

{ int num；

if(r==NULL) num=0；

else

if(r->firstchild==NULL)

num= +numberofleaf；

(r->nextbrother)；

else ；

return(num)；

}

2．在根结点为r的二叉排序树中，插人数据域值为x的结点，要求插入新结点后的树仍是一棵二叉排序树(NULL代表空指针)。

二叉排序树的结点结构为

typedef struct node

{ int key；

struct node \*lc,\*rc；

}BiNode；

BiNode \*insert(BiNode \*r,int x)

{ BiNode \*p,\*q,\*s;

s=(BiNode\*)malloc(sizeof(BiNode));

s->key=x;

s->lc=NULL;s->rc=NULL;

q=NULL;

if(r==NULL) {r=s;return(r);}

p=r;

while( )

{ q=p;

if( ) p=p->lc;

else p=p->rc

}

if(x<q->key) ;

else ;

return;

}

清华大学2001年数据结构与程序设计试题

试题内容：

一、试给出下列有关并查集(mfsets)的操作序列的运算结果：

union(1，2)，union(3，4)，union(3，5)，union(1，7)，union(3，6)，

union(8，9)，union(1，8)，union(3，10)，union(3，11)，union(3，12)，

union(3，13)，union(14，15)，union(16，0)，union(14，16)，union(1，3)，

union(1，14)。(union是合并运算，在以前的书中命名为merge)

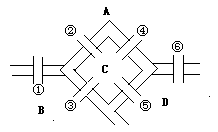
要求：

(1)对于union(i，j)，以i作为j的双亲；(5分)

(2)按i和j为根的树的高度实现union(i，j)，高度大者为高度小者的双亲；(5分)

(3)按i和j为根的树的结点个数实现union(i，j)，结点个数大者为结点个数小者的双亲。(5分)

二、设在4地(A，B，C，D)之间架设有6座桥，如下图所示：



要求从某一地出发，经过每座桥恰巧一次，最后仍回到原地。

(1)试就以上图形说明：此问题有解的条件是什么?(5分)

(2)设图中的顶点数为n，试用C或Pascal描述与求解此问题有关的数据结构并编写一个算法，找出满足要求的一条回路。(10分)

三、针对以下情况确定非递归的归并排序的运行时间(数据比较次数与移动次数)：

(1)输人的n个数据全部有序；(5分)

(2)输入的n个数据全部逆向有序；(5分)

(3)随机地输入几个数据。(5分)

四、简单回答有关AVL树的问题。

(1)在有N个结点的AVL树中，为结点增加一个存放结点高度的数据成员，那么每一个结点需要增加多少个字位(bit)?(5分)

(2)若每一个结点中的高度计数器有8bit，那么这样的AVL树可以有多少层?最少有多少个关键码?(5分)

五、一个散列表包含hashSize=13个表项，其下标从0到12，采用线性探查法解决冲突。请按以下要求，将下列关键码散列到表中。

10 100 32 45 58 126 3 29 200 400 0

(1)散列函数采用除留余数法，用％hashSize(取余运算)将各关键码映像到表中。请指出每一个产生冲突的关键码可能产生多少次冲突。(7分)

(2)散列函数采用先将关键码各位数字折叠相加，再用％hashSize将相加的结果映像到表中的办法。请指出每一个产生冲突的关键码可能产生多少次冲突。(8分)

六、设一棵二叉树的结点定义为

struct BinTreeNode{

ElemType data；

BinTreeNode \*leftChild,\*rightChild；

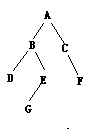
}

现采用输入广义表表示建立二叉树。具体规定如下：

(1)树的根结点作为由子树构成的表的表名放在表的最前面；

(2)每个结点的左子树和右子树用逗号隔开。若仅有右子树，没有左子树，逗号不能省略。

(3)在整个广义表表示输入的结尾加上一个特殊的符号(例如“#”)表示输入结果。；

例如，对于如右图所示的二叉树，其广义表表示为A(B(D，E(G，))，C(，F))

此算法的基本思路是：依次从保存广义表的字符串ls中输入每个字符。若遇到的是字母(假定以字母作为结点的值)，则表示是结点的值，应为它建立一个新的结点，并把该结点作为左子女当(k=1)或右子女(当k=2)链接到其双亲结点上。若遇到的是左括号”(”，则表明子表的开始，将A置为1；若遇到的是右括号”)”，则表明子表结束。若遇到的是逗号”，”，则表示以左子女为根的子树处理完毕，应接着处理以右子女为根的子树，将A置为2。

在算法中使用了一个栈s，在进入子表之前，将根结点指针进栈，以便括号内的子女链接之用。在子表处理结束时退栈。相关的栈操作如下：

MakeEmpty(s)置空栈

Push(s，p)元素p进栈

Pop(s)退栈

Top(s)存取栈顶元素的函数

下面给出了建立二叉树的算法，其中有5个语句缺失。请阅读此算法并把缺失的语句补上。(每空3分)

Void CreateBinTree(BinTreeNode \*&BT,char ls)

{ Stack<BinTreeNode\*>s；MakeEmpty(s)；

BT=NULL； //置二叉树

BinTreeNode \*p；

int k；

istream ins(ls)； //把串ls定义为输入字符串流对象ins

char ch；

ins>>ch； //从ins顺序读入一个字符

while(ch!=”#”)

{ //逐个字符处理，直到遇到’#’为止

switch(ch)

{ case’(’： (1)

k=1；

break；

case ’)’：pop(s)；

break；

case ‘,’ (2)

break；

default：p=new BinTreeNode；

(3)

p->leftChild=NULL；

p->rightChfild=NULL；

if(BT==NULL)

(4)

else if(k==1) top(s)->leftChild=p；

else top(8)->rightChild=p；

}

(5)

}

}|

七、下面是一个用C编写的快速排序算法。为了避免最坏情况，取基准记录pivot采用从left，right和mid=⎣ (1eft+right)/2⎦中取中间值，并交换到right位置的办法。数组a存放待排序的一组记录，数据类型为Type，left和right是待排序子区间的最左端点和最右端点。

Void quicksort(Type a&#, int lefl，int right)

{ Type temp；

if(left<right)

{ Type pivot=medlian3(a,left，right)；

int i=left，j=right-1;

for( ； ；)

{ while(i<j&&s[I]<pivot) i++；

while(i<j&&pivot<a[j] j--；

if(i<j)

{ temp=a[i]；a[j]=a[i]；a[i]=temp；

i++；j--；

}

else break；

}

if(s[i]>pivot)

{ temp=a[i]；a[i]=a[right]；a[right]=temp；}

quicksort(a,left，i-1)； //递归排序左子区间

quicksort(a，i+1，right)；//递归排序右子区间

}

}

(1)用C或Pascal实现三者取中子程序median3(a,left,right)；(5分)

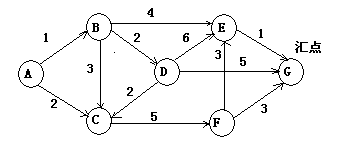
(2)改写quicksoft算法，不用栈消去第二个递归调用quicksort(a，j+1，right)；(5分)

(3)继续改写quicksort算法，用栈消去剩下的递归调用。(5分)

上海交通大学2001年数据结构及程序设计试题

注意：程序设计请采用C语言，程序应有注解及说明。回答问题简洁、清晰全面。不得采用类C之类的语言写程序。

一、参见下图，该有向图是AOE网络。该图上已标出了源点及汇点，并给出了活动(边)的权值。



请求出该AOE网络的关键路径，以及事件(结点)的最早发生时间及最迟发生时间。(本题8分)

二、已知某二叉树的每个结点，要么其左、右子树皆为空，要么其左、右子树皆不空。又知该二叉树的前序序列为(即先根次序)：J、F、D、B、A、C、E、H、X、I、K；后序序列为(即后根次序)：A、C、B、E、D、X、I、H、F、K、Jo请给出该二叉树的中序序列(即中根次序)，并画出相应的二叉树树形。(本题8分)

三、回答下列问题：(本题10分)

1)具有N个结点且互不相似的二叉树的总数是多少?

2)具有N个结点且不同形态的树的总数是多少?

3)对二叉树而言，如果它的叶子结点总数为N0，度为2的结点的总为N2，则N0和N2之间的关系如何?

4)二叉树是否是结点的度最多为2的树?请说明理由。

5)具有n片叶子的哈夫曼树(即赫夫曼树)中，结点总数为多少？

四、在外部分类时，为了减少读、写的次数，可以采用k路平衡归并的最佳归并树模式。当初始归并段的总数不足时，可以增加长度为零的“虚段”。请问增加的“虚段”的数目为多少?请推导之。设初始归并段的总数为m。(本题8分)

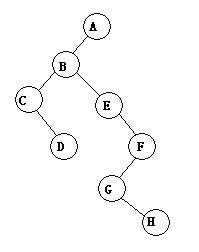
五、对平衡的排序二叉树进行删除结点的操作，必须保证删除之后平衡树中的每个结点的平衡因子是+1，-1，0三种情况之一，且该树仍然是排序二叉树。现假定删除操作是在p结点的左子树上进行的，且该左子树原高为h-l，现变为h-2。因此，必须从p的左子树沿着到根的方向回溯调整结点的平衡因子，并进行树形的调整。设p是调整时遇到的第一个平衡因子力图由-1变成-2的最年轻的“前辈”结点。我们知道，以p为根的子树经调整后，高度有可能减少1。试用图形把调整前及调整后的相关结点的平衡因子、树形表示出来；仅仅针对调整后子树的高度减少1的情况即可。注意，罗列出所有可能的情况。上图可供参考。(本题10分)

六、某算法由下述方程表示。请求出该算法的时间复杂性的级别(以大O形式表示)。注意n>7求解问题的规模，设n是3的正整数幂。(本题8分)

1 如果n=1

5T(n/3)+n 如果n>1

Tn=

七、如右图所示，该二叉树是某棵有序树变换成的相对应的二叉树。试给出原来的有序树的形状。并回答以下问题：

1)原有序树是度为多少的树?

2)原有序树的叶子结点有哪几个?

3)是否所有的二叉树都可以找到相对应的有序树?必须满足什么条件?(本题5分)

八、在排序二叉树上进行查找操作时，设对树中的每个结点查找概率相同。设由n个结点构成的序列生成的排序二叉树是“随机”的。试求出在成功查找的情况下，平均查找长度是多少?为了简单起见，最后得到的递推式可不予求解。(本题8分)

九、设从键盘每次输入两个字符。如A、B，则表示存在着一条由数据场之值为字符A的结点到数据场之值为字符B的结点的有向边。依此输入这些

有向边，直至出现字符!为止。试设计一个程序，生成该有向图的邻接表及逆邻接表。必须交待所用的结构、变量、加以适当注解。(本题20分) ·

十、设二叉树中结点的数据场之值为一字符。采用二叉链表的方式存储该二叉树中的所有结点，设p为指向树根结点的指针。设计一个程序在该二叉树中寻找数据场之值为key(key为一变量，变量内容为一字符)的那个结点的所有祖先。设二叉树中结点数据场之值互不重复。(本题15分)

注意：有些书上将二叉树的二叉链表存储形式称之为标准存储形式。

南开大学2001年数据结构试题

一、选择题(每小题3分，共21分)

在下列各题中，每题之后均有若干个备选答案，请选出所有正确的答案，填人“ ”处。答案请写在答题纸上。

1．任何一个无向连通图的最小生成树 。

①只有一棵 ②有一棵或多棵

③一定有多棵 ④可能不存在

2．已知一棵非空二叉树的 ，则能够惟一确定这棵二叉树。

①先序遍历序列和后序遍历序列

②先序遍历序列和中序遍历序列

③先序遍历序列

④中序遍历序列 ，

3．使用指针实现二叉树时，如果结点的个数为n，则非空的指针域个数为 。

①n-1 ②2n-1 ③n+1 ④2n+1

4．设队列存储于一个一维数组中，数组下标范围是1-n，头尾指针分别为f和r，f指向第一个元素的前一个位置，r指向队列中的最后一个元素，则队列中元素个数为 。

①r-f ②r-f+1 ③(r-f+1)mod n ④(r-f+n)mod n

5．任意一个AOE网络的关键路径 。

①一定有多条 ②只有一条 ③可能不只一条 ④可能不存在 ．

6．下列排序算法中，在每一趟都能选出一个元素放到其最终位置上的是 。

①插入排序 ②希尔排序 ③快速排序 ④堆排序

7．任意一个有向图的拓扑排序序列 。

①一定有多种 ②只有一种 ③可能不存在 ④以上都不对

二、(7分)已知散列表地址空间为0-8，哈希函数为H(k)=kmod7，采用线性探测再散列法解决冲突。将下面关键字数据依次填入该散列表中，同时将查找每个关键字所需的比较次数m填入下表中，并求等概率下的平均查找长度。

关键字值：100，20，21，35，3，78，99，45

0 1 2 3 4 5 6 7 8

A

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 100 | 20 | 21 | 35 | 3 | 78 | 99 | 45 |
|  |  |  |  |  |  |  |  |

m

三、(12分)回答下列问题：

(1)(3分)什么叫基数排序?

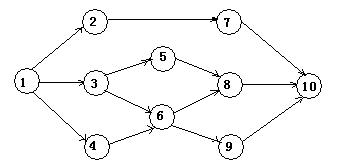
(2)(3分)什么是AVL树中的平衡因子，它有什么特点?

(3)(6分)n个元素的序列{k1,k2,…kn}满足什么条件才能称之为堆?简述对它进行堆排序的过程。

四、(10分)顺序给出以下关键字：63、23、31、26、7、91、53、15、72、52、49、68，构造3阶B-树。要求从空树开始，每插入一个关键字，画出一个树形。

五、(6分)设有向无环图G如下图所示：

试写出图G的六种不同的拓扑排序序列。



六、(10分)设二叉树以二叉链表表示，各结点的结构如下所示：

|  |  |  |  |
| --- | --- | --- | --- |
| left | data | subsum | right |

其中left、right分别为指向该结点左、右孩子的指针，data为存储关键字值的整数域，subsum中存储以该结点为根的子树中所有关键字值之和。试使用C或C++语言设计算法，计算所给树T中所有结点的subsum值。

七、(12分)给出一个带表头的单链表L，L的每个结点中存放一个整数。现给定一个阈值K，将L分成两个子表L1和L2，其中L1中存放L中所有关键字值大于等于是的结点，L2中存放L中所有关键字值小于K的结点。试编程实现这个过程。要求，使用C或C++语言实现算法，L1和L2仍占用L的存储空间。

八、(10分)设有一维整数数组A，试使用C或C++语言设计算法，将A中所有的奇数排在所有偶数之前。要求，时间复杂度尽可能地少，结果仍放在A原来的存储空间。

九、(12分)简述哈夫曼编码的构造过程。

华东理工大学2001年数据结构与程序设计试题

一、单选题(10分)

1．若某线性表中最常用的操作是在最后一个元素之后插入一个元素和删除第一个元素，则采用——存储方式最节省运算时间。

A．单链表 B．仅有头指针的单循环链表

C．双链表 D．仅有尾指针的单循环链表

2．若在线性表中采用折半查找法查找元素，该线性表应该 。

A．元素按值有序 B．采用顺序存储结构

C．元素按值有序，且采用顺序存储结构

D．元素按值有序，且采用链式存储结构

3．对于给定的结点序列abcdef，规定进栈只能从序列的左端开始。通过栈的操作，能得到的序列为 。

A．abcfed B．cabfed C．abcfde D．cbafde

4．已知一算术表达式的中缀形式为A+B\*C-D/E，后缀形式为ABC\*+DE/-，其前缀形式为 。

A．-A+B\*C/DE B．-A+B\*CD/E

C．-+\*ABC/DE D．-+A\*BC/DE

5．如果n1和n2是二叉树T中两个不同结点，n2为n1的后代，那么按遍历二叉树T时，结点n2一定比结点n1先被访问。

A．前序 B．中序 C．后序 D．层次序

6．具有65个结点的完全二叉树其深度为 。(根的层次号为1)

A．8 B．7 C．6 D．5

7．已知二叉树的前序遍历序列是abdgcefh，中序遍历序列是dgbaechf，它的后序遍历序列是 。

A．bdgcefha B. gdbecfha C．bdgechfa D．gdbehfca

8．对于前序遍历和后序遍历结果相同的二叉树为 。

A．一般二叉树

B．只有根结点的二叉树

C．根结点无左孩子的二叉树

D．根结点无右孩子的二叉树

9．对于前序遍历与中序遍历结果相同的二叉树为 。

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

10．在有n个叶子的哈夫曼树中，其结点总数为 。

A．不确定 B．2n C．2n+1 D．2n-1

二、是非题(10分)

1．顺序存储方式只能用于存储线性结构。

2．消除递归不一定需要使用栈。

3．将一棵树转换成相应的二叉树后，根结点没有右子树。

4．完全二叉树可以采用顺序存储结构实现，存储非完全二叉树则不能。

5．在前序遍历二叉树的序列中，任何结点的子树的所有结点都是直接跟在该结点之后。

6．邻接表法只能用于有向图的存储，而邻接矩阵法对于有向图和无向图的存储都适用。

7．在一个有向图的邻接表中，如果某个顶点的链表为空，则该顶点的度一定为零。

8．二叉树为二叉排序树的充分必要条件是其任一结点的值均大于其左孩子的值，小于其右孩子的值。

9．最佳二叉排序树的任何子树都是最佳二叉排序树。

10．对两棵具有相同关键字集合的而形状不同的二叉排序树，按中序遍历它们得到的序列的顺序是一样的。

三、问答题

(应届生限做2，3，4，5题；在职生任选做四题；共40分)

1．(10分)什么是广义表?请简述广义表与线性表的主要区别。

2．(10分)下图表示一个地区的通讯网，边表示城市间的通讯线路，边上的权表示架设线路花费的代价。

①请分别给出该图的邻接表和邻接矩阵，要求每种存储结构能够表达出该图的全部信息，并分别对这二种形式中每个部分的含义(物理意义)予以简要说明。

②若假设每个域(包括指针域)的长度为一个字节，请分别计算出这二种结构所占用的空间大小。

3．(10分)已知如下所示长度为12的表：

(Jan，Feb，Mar，Apr，May，June，July，Aug，Sep，Oct，Nov，Dec)

①试按表中元素的顺序依次插入一棵初始为空的二叉排序树，请画出插入完成之后的二叉排序树，并求其在等概率的情况下查找成功的平均查找长度。

②若对表中元素先进行排序构成有序表，求在等概率的情况下对此有序表进行折半查找时查找成功的平均查找长度。

③按表中元素顺序构造一棵平衡二叉排序树，并求其在等概率的情况下查找成功的平均查找长度。

4．(10分)在起泡排序过程中，有的关键字在某趟排序中可能朝着与最终排序相反的方向移动，试举例说明之。快速排序过程中有没有这种现象?

5．(10分)调用下列函数f(n)，回答下列问题：

①试指出f(n)值的大小，并写出f(n)值的推导过程；

②假定n=5，试指出f(5)值的大小和执行f(5)的输出结果。

C函数

int f(int n)

{ int i，j，k，sum=0；

for(j=1；i<n+1 i++)

{ for(j=n；j>i-1；j--)

for(k=1；k<j+1；k++)

suni++；

prinft(”sum=%d\n”，sum)；

}

return(sum)；

}

Pascal函数

FUNCTION f(n：integer)：integer；

VAR i，j，k，sum，integer；

BEGIN ．

sum：=0；

FOR i：=1 TO n DO

BEGIN

FOR j：=n DOWN i DO

FOR k：=1 TO j DO

sum：=sum+1

writeln(’sum=’，sum)

END；

f：=sum

END；

四、编写算法

(应届生限做1、2、3、4题，在职生任选四题，每题10分，共40分)

1．(10分)利用两个栈S1和S2模拟一个队列，写出入队和出队的算法(可用栈的基本操作)。

栈的操作有：

makeEmpty(s：stack)； 置空栈

push(s：stack；value：datatype)； 新元素value进栈

pop(s：stack)：datatype； 出栈，返回栈顶值

isEmpty(s：stack)：boolean； 判栈空否

队列的操作有

enQtleue(s1：stack；s2：stack；value：datatype)；元素value进队

deQueue(s1：stack；s2：stack)：datatype； 出队列，返回队头值

2．(10分)试写出给定的二叉树进行层次遍历的算法，在遍历时使用一个链接队列。

3．(10分)试写一个算法，判别以邻接表方式存储的有向图中是否存在由顶点vi到顶点vj的路径(i<>j)。假设分别基于下述策略：

1)图的深度优先搜索；

2)图的广度优先搜索；

4．(10分)设二叉排序树中关键字由1至1000的整数构成，现要检索关键字为531的结点，下述关键字序列中哪些可能是二叉排序树上搜索到的序列，哪些不可能是二叉排序树上搜索到的序列，哪些不可能是二叉排序树上搜索到的序列？

a)800，399，588，570，500，520，566，531

b)730，355，780，390，701，401，530，531

c)732，321，712，385，713，392，531

d)8，578，555，340，433，551，550，450，531

通过对上述序列的分析，试写一个算法判定给定的关键字序列(假定关键字互不相同)是否可能是二叉排序树的搜索序列。若可能则返回真，否则返回假。可假定被判定的序列已存入数组中。

5．(10分)编写一个在非空的带表头结点的单链表上实现就地排序的程序。(不额外申请新的链表空间)

华中理工大学2001年数据结构试题

说明：在有数据类型定义和算法设计的各题中，任取C语言、PASCAL语言之一作答，但不许使用所谓“类C”或“类PASCAL”。

一、选择题(从下列各题四个备选答案中选出一至四个正确答案，将其代号(A，B，C，D)写在题干前面的括号内。答案选错或未选全者，该题不得分。每小题1分，共计20分)

( )1．一个算法具有 等特点。

A．可行性 B．至少有一个输入量

C．确定性 D．健壮性

( )2. 是一个线性表。

A．(A，B，C，D) B．{’A’，’B’，’C’，’D’}

C．(1，2，3，…) D．(40，-22，88)

( )3．可使用 作压缩稀疏矩阵的存储结构。

A．邻接矩阵 B．三元组表

C．邻接表 D．十字链表

( )4．采用一维数组表示顺序存储结构时，可用它来表示 。

A．字符串 B．2度树 C．二叉树 D．无向图

( )5．假设进入栈的元素序列为d，c，a，b，e，那么可得到出栈的元素序列 。

A．a，b，c，d，e B．a，b，e，d，c

C．d，b，a，e，c D．d，b，a，c，e

( )6．对队列的操作有 。

A．在队首插入元素 B．删除值最小的元素

C．按元素的大小排序 D．判断是否还有元素

( )7．假设有60行70列的二维数组a[1．．60，1．．70]，以列序为主序顺序存储，其基地址为10000，每个元素占2个存储单元，那么第32行第58列的元素a[32，58]的存储地址为 。(无第0行第0列元素)

A．16902 B．16904

C．14454 D．答案A，B，C均不对

( )8. 是C语言中"abcd321ABCD"的子串。

A．abed B．32lAB C．”abeABC” D．”2lAB”

( )9．在下列各广义表中，长度为3的广义表有 。

A．(a，b，c，()) B．((g)，(a，b，c，d，f)，())

C．(a，(b，(c))) D．((()))

( )10．一个堆(heap)又是一棵 。

A．二叉排序树 B．完全二叉树

C．平衡二叉树 D．哈夫曼(Huffman)树

( )11．折半查找有序表(4，6，10，20，30，50，70，88，100)，若查找元素58，则它将依次与表中元素 比较大小，查找结果是失败。

A．20，70，30，50 B．30，88，70

C．20，50 D．30，88，50，70

( )12．对27个记录的有序表作折半查找，当查找失败时，至少需要比较 次关键字。

A．3 B．4 C．5 D．6

( )13．具有12个结点的完全二叉树有 。

A．5个叶子 B．5个度为2的结点

C．7个分支结点 D．2个度为1的结点

( )14．具有n(n>0)个结点的完全二叉树的深度为 。

A．⎡log2(n)⎤ B．⎣log2(n+1)⎦

C．⎣log2(n)⎦+1 D．⎣loge(n) +1⎦

( )15．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 。

A．32 B．33 C．34 D．15

( )16．对14个记录的表进行2路归并排序，共需移动 次记录。

A．42 B．56 C．91 D．84

( )17．对有n个记录的表作快速排序，在最坏情况下，算法的时间复杂度是 。

A．O(n) B．O(n2) C．O(nlog2n) D．O(2n)

( )18．在最好情况下，下列算法中 排序算法所需比较关键字次数最少。

A．冒泡 B．归并 C．快速 D．直接插入

( )19．置换选择排序的功能是 。

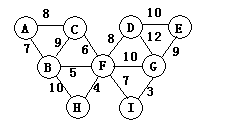
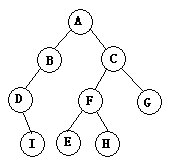
A．选出最大的元素 B．产生初始归并段

C．产生有序文件 D．置换某个记录

( )20．可在 构造一个散列文件。

A．内存 B．软盘 C．磁带 D．硬盘

二、试用2种不同表示法画出下列二叉树B1的存储结构，并评述这2种表示法的优、缺点。(共10分)



二题图B1 三题图G

三、对图G：

1. 从顶点A出发，求图G的一棵深度优先生成树；
2. 从顶点B出发，求图G的一棵广度优先生成树；

3．求图G的一棵最小生成树。(共12分)

四、设哈希(Hash)函数为：H(k)=kMODl4，其中k为关键字(整数)，MOD为取模运算，用线性探测再散列法处理冲突，在地址范围为0~14散列区中，试用关键字序列(19，27，26，28，29，40，64，21，15，12)造一个哈希表，回答下列问题：

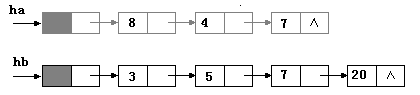
1．画出该哈希表的存储结构图；

2．若查找关键字40，必须依次与表中哪些关键字比较大小？

3．假定每个关键字的查找概率相同，试求查找成功时的平均查找长度。(共13分)

五、给定头指针为ha的单链表A，和头指针为hb的递增有序单链表B。试利用表A和表B的结点，将表A和表B归并为递增有序表C，其头指针为hc，允许有相同data值(数据元素)的结点存在，表A，B，C均带表头结点。

例如，给定：



将它们归并为：



1．在下列C算法merge中有下划线的位置填空，使之成为完整的算法，要求在填空后加注释，以提高算法的可读性。

2．假定单链表A和B的长度分别为m和n(m>0，n>0)，试分别就最好情况、最坏情况和平均性能，分析算法merge的时间复杂度。(共15分)

结点类型定义如下：

struct node 、

{ int data；

struct node \*next；

}；

算法如下：

Void merge(struct node\*ha，struct node \*hb，struct node\*hc)

{ struct node \*pc，\*qc，\*pa，\*fa；

int search；

hc=hb； hb=NULL；

pa=ha->next； free(ha)；

while(pa)

{pc=hc； qc=hc->next；

search= ；

while(qc&&search)

{if(pa->data<=qc->data)

；

else

{ pc=qc；

qc= ；

}

}

fa= ；

pa= ；

fa->next= ；

；

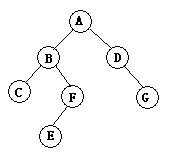
}

}

六、设下图二叉树B2的存储结构为二叉链表，root为根指针，结点结构为：(1child，data，rchild)其中：lchild，rchild分别为指向左右孩子的指针，data为字符型。(共10分)

1. 对下列二叉树B2，执行下列算法traversel(root)，试指出其输出结果；

2．假定二叉树B2共有n个结点，试分析算法traversal的时间复杂度。结点类型定义如下：

 struct node

{ chardata；

struct node \*lchild，\*rchild；

}；

算法如下：

void traversal(struct node \*root)

{ if(root)

{ prinft(“%c”，root->data)；

traversal(root->lchild)；

二叉树B2

prinft(”%c”，root->data)；

traversal(root->rchild)；

}

}

七、算法设计题(要求：(1)写出所用数据结构的类型定义和变量说明；(2)写出算法，并在相关位置加注释，以提高算法的可读性。)

1．试设计一算法：输入一个有m行n列的整数矩阵，然后将每一行的元素按非减次序输出。例如，若输入：

4，3，5，6，2

9，8，1，2，8

7，1，2，3，8

则输出如下结果：

2，3，4，5，6

1，2，8，8，9

1，2，3，7，8

2．如果字符串的一个子串(其长度大于1)的各个字符均相同，则称之为等值子串。试设计一算法：输入字符串S，以’!’为结束标志，如果串S中不存在等值子串，则输出信息：”无等值子串”，否则求出(输出)一个长度最大的等值子串。

例如，若S=”abcl23abcl23！”，则输出：”无等值子串”；

又如，若S=”abcaabcccdddddaaadd!”，则输出等值子串：”ddddd”。(共20分)

北京理工大学2001年程序设计(含数据结构)试题

第二部分 数据结构(共50分)

一、选择题(12分，每题2分)

1．下列数据中哪些是非线性数据结构 。

A．栈 B．队列 C．完全二叉树 D．堆

2．静态链表中指针表示的是 。

A．内存地址 B．数组下标

C．下一元素地址 D．左、右孩子地址

3．用不带头结点的单链表存储队列时，其队头指针指向队头结点，其队尾指针指向队尾结点，则在进行删除操作时 。

A．仅修改队头指针

B．仅修改队尾指针

C．队头，队尾指针都要修改

D．队头，队尾指针都可能要修改

4．在下列排序算法中，哪一个算法的时间复杂度与初始排列无关 。

A．直接插入排序 B．起泡排序

C．快速排序 D．直接选择排序

5．二叉树第i层结点的结点个数最多是(设根的层数为1) 。

A．2i-1  B．2i-1 C．2i D．2i-1

6．树的后根遍历序列等同于该树对应的二叉树的 。

A．先序序列 B．中序序列 C．后序序列

二、填空(8分，每空1分)

1．数据结构中评价算法的两个重要指标是——。

2．顺序存储结构是通过 表示元素之间的关系的。链式存储结构是通过 表示元素之间的关系的。

3．AOV网中，结点表示 ，边表示 。AOE网中，结点表示 ，边表示 。

4．哈夫曼树是 。

三、求解下列问题(25分，每题5分)

1．请用C语言给出线性链表的类型定义。

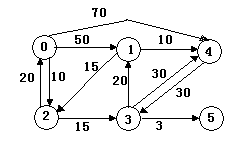
2．已知二叉树的先序序列：CBHEGAF，中序序列：HBGEACF，试构造该二叉树。

3．设散列表的地址空间为0到12的存储单元，散列函数为：h(key)=key mod 13，用链地址法解决冲突，初始时散列表为空。现依次将关键字25，33，48，25，43，38，39插入散列表，试画出完成上述所有插入操作后散列表的状态，并计算在等概率的情况下，在该表中查找成功的平均查找长度。

4．给出如下图所示的图形。

1)试写出该图的邻接表；

2)试写出该图从结点0出发的深度优先遍历序列和广度优先遍历序列，并画出遍历过程中所经的路径。



5．全国统考答题

对上图，按迪杰斯特拉算法求出从结点0到其余各点的最短路径，并给出在求解过程中数组distance的变化状态。

6．单独考试答题(可在5或6中任选一题做)

给出关键字序列27，18，21，77，26，45，66，34试写出快速排序的过程。

四、算法题(12分)(请在算法的主要步骤上加注释)

1．下面是用C语言编写的对不带头结点的单链表进行就地置逆的算法，该算法用L返回置逆后链表的头指针。试在空缺处填入适当的语句。

void reverse(1inklist&L)

{ p=NULL；q=L；

while(q!=NULL)

{

q->next=p；

p=q；

}

}

2．全国统考答题

设二叉树用二叉链表存储，试编写按层输出二叉树结点的算法。

3．单独考试答题(可在2或3中任选一题做)

试编写在带头结点的单链表中删除(一个)最小值结点的(高效)算法。

void delete(Linklist &L)

北京邮电大学2001年数据结构试题

一、选择题(10分，每题2分)

1．B+树是应用在( )文件系统中。

①ISAM ②VSAM

2．将一棵树t转换为孩子-兄弟链表表示的二叉树h，则t的后根遍历是h的( )。

①前序遍历 ②中序遍历 ③后序遍历

3．一个有向无环图的拓扑排序序列( )是惟一的。

①一定 ②不一定

4．将10个元素散列到100000个单元的哈希表中，则( )产生冲突。

①一定会 ②一定不会 ③仍可能会

5．若要求尽可能快地对序列进行稳定的排序，则应选( )。

①快速排序 ②归并排序 ③冒泡排序

二、填空题(20分，每空2分)

1．数据的逻辑结构是指 。

2．区分循环队列的满与空，只有两类办法，它们是 和 。

3．用一维数组B以列优先存放带状矩阵A中的非零元素A[i，j](1≤i≤n，i-2≤j≤i+2)，B中的第8个元素是A中第 行、第 列的元素。

4．字符串’ababaaab’的nextval函数值为——。

5．下图中的强连通分量的个数为 个。

6．外部排序的基本方法是归并排序，但在之前必须先生成 。

7．若不考虑基数排序，则在排序过程中，主要进行的两种基本操作是关键字的 和记录的

三、简答题(15分，每题5分)

1．特殊矩阵和稀疏矩阵哪种压缩存储后会失去随机存取的功能?为什么?

2．试问线索二叉树的目的何在?

3．在多关键字排序时，LSD和MSD两种方法的特点是什么?

四、应用题(25分，每题5分)

1．画出按下表中元素的顺序构造的平衡二叉排序树，并求其在等概率的情况下查找成功的平均查找长度。

{15，12，24，3，27，21，18，6，36，33，30，26，3}

2．假设用于通信的电文由字符集{a，b，c，d，e，f，g}中的字母构成，它们在电文中出现的频率分别为{0.31，0.16，0.10，0.08，0.11，0.20，0.04}。

(1)为这7个字母设计哈夫曼编码；

(2)对这7个字母进行等长编码，至少需要几位二进制数?哈夫曼编码比等长编码使电文总长压缩多少?

3．试推导当总盘数为n时的Hanoi塔的移动次数。

4．一棵满k叉树，按层次遍历存储在一维数组中，试计算结点下标为u的结点的第i个孩子的下标以及结点下标为v的结点的父母结点的下标。

5．有一图的邻接矩阵如下，试给出用弗洛伊德算法求各点间最短距离的矩阵序列A(1)、A(2)、A(3)和A(4)。

A=

0 2 ∞ ∞

∞ 0 1 6

5 ∞ 0 4

3 ∞ ∞ 0

五、算法(30分，每题10分)

1．在单链表中，每个结点含有5个正整型的数据元素(若最后一个结点的数据元素不满5个，以值0填充)，试编写一算法查找值为n(n>0)的数据元素所在的结点指针以及在该结点中的序号，若链表中不存在该数据元素则返回空指针。

2．将一组数据元素按哈希函数H(key)散列到哈希表m(0．．m)中，用线性探测法处理冲突(H(key)+1、H(key)+2、…、H(key)-1，假设空单元用EMPTY表示，删除操作是将哈希表中结点标志位从INUSE标记为DELETED，试写出该散列表的查找、插入和删除三个基本操作算法。

3．给出算法将二叉链表表示的表达式二叉树按中缀表达式输出，并加上相应的括号。

北京航空航天大学2001年 数据结构与程序设计试题

一、问答题(本题10分)

一般情况下，线性表可以采用哪几种存储结构?请分别叙述每一种存储结构的构造原理与特点。

二、(本题10分)

已知AOE网为G=(V，E)，V={v1，v2，v3，v4，v5，v6，v7，v8，V9，vl0}，E={a1，a2，a3，a4，a5，a6，a7，a8，a9，al0，a11，a12，a13，a14}，其中：

a1：(v1，v2)5 a2：(v1，v3)6 a3：(v2，v5)3 a4：(v3，V4)3

a5：(v3，v5)6 a6：(v4，v5)3 a7：(v4，v7)1 a8：(v4，v8)4

a9：(v5，v6)4 al0：(v5，v7)1 a11：(v6，vl0)4 a12：(v7，v9)5

a13：(v8，v9)2 a14：(v9，v10)2

注：顶点偶对右下角的数字表示边上的权值。

请按下述过程指出所有关键路径：

ee[1:10]:

le[1:10]:

e[1:14]:

l[1:14]:

其中，ee[i]与le[i]分别表示事件vi的最早发生时间与最晚发生时间；e[i]与l[i]分别表示活动ai的最早开始时间与最晚开始时间。

三、(本题共30分，每小题10分)

欲建立一文献库，其正文(文献本身)存放在一个双向循环链表的各个链结点中。

1．为便于链结点的插入、删除操作，以及按题目、发表日期、发表者名称、主题词(假设每文最多给出三个主题词)进行检索，请设计该链表的链结点结构(给出链结点每个域的名称，并说明该域内存放什么信息。注：以下各小题设计链结点结构也这样要求)。画出整个链表结构的示意图。

2．设计一个三级索引结构，其中第三级索引称为题目索引，是按文献题目构造的稠密索引，通过该级索引并根据给定题目可得到每个文献的存放地址；该级索引按文献学科类分类存放。第二级索引称为中类索引，是题目索引的索引，指出同一中类的文献题目索引的存放位置(例如农林类、气象类……，古代史类、近代史类……)。第一级索引称为大类索引，指出同一大类(如：自然科学类、历史类……)的文献的中类索引的存放位置。请设计每一级索引的结点结构，并画出该索引的整体示意图。

3．再设计一种三级索引结构，其中第三级索引仍是题目索引(与2题所述相同)，第二级索引把具有相同主题词的文献题目索引地址组织在一个单链表中。第一级索引称为主题词索引，用文献给出的主题词作关键字组成杂凑表，即该级索引为一个杂凑表，能够指出具有同一主题词的文献题目索引的索引链表的第一个链结点的存储位置。该杂凑表采用链地址法处理冲突。请设计每一

级索引的结点结构，并画出该索引的整体示意图。

四、(本题10分)

已知非空线性链表由list指出，链结点的构造为

|  |  |
| --- | --- |
| data | link |

请写一算法，将链表中数据域值最小的那个链结点移至链表的最前面。要求：不得额外申请新的链结点。

五、(本题15分，第1小题5分，第2小题10分)

已知求两个正整数m与n的最大公因子的过程用自然语言可以表述为反复执行如下动作：

第一步：若n等于零，则返回m；

第二步：若m小于n，则m与n相互交换；否则，保存m，然后将n送m，将保存的m除以n的余数送n。

1．将上述过程用递归函数表达出来(设求x除以y的余数可以用xMODy形式表示)。

2．写出求解该递归函数的非递归算法。

六、(本题10分)

函数void insert(char \*s，char \*t，int pos)将字符串t插入到字符串s中，插入位置为pos。请用C语言实现该函数。假设分配给字符串s的空间足够认字符串t插入。(说明：不得使用任何库函数。)

七、(本题15分) 。

命令sgrep用来在文件中查找给定字符串，并输出串所在行及行号。

命令格式为：sgrep[-i]filename searchstring

其中：-i：表示查找的大小写无关，省略时表示大小写相关。

filename：给定文件名。

searchstring：所要查找的串。

用C语言实现该程序，该程序应具有一定的错误处理能力。(提示：使用命令行参数)

注意：除文件及输入/出操作可使用库函数外，其他不允许使用库函数。

西安交通大学2001年数据结构试题

一、判断下列叙述是否正确，正确的填√，不正确的填×(10分)

1．数据对象就是一组数据元素的集合。 ( )

2．任何一棵前序线索二叉树，都可以不用栈实现前序遍历。 ( )

3．就平均查找长度而言，分块查找最小，折半查找次之，顺序查找最大。 ( )

4．用shell方法排序时，若关键字的排列杂乱无序，则效率最高。 ( )

5．在m阶B-树中，所有非终端结点至少包含⎡m／2⎤。 ( )

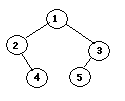
二、选择填空题(15分)

1．假设以数组A[m．．n]存放循环队列的元素，其头指针是front，当前队列有k个元素，则队列的尾指针为( )。

A．(front+k)mod(n-m+1)

B．(m+k)mod n+front

C．(front-m+k)mod(n-m+1)+m

 D．(front-m+k)mod(n-m+1)

1. 已知二叉树如右图所示，此二叉树的顺

序存储的结点序列是( )。

A．123□45 B．12345

C．12□435 D．□24153

3．若用冒泡排序对关键字序列{20，17，11，8，6，2}从小到大进行排序，则需要交换的总次数为( )。

A．3 B．6 C．12 D．15

4．对于一个头指针为head的带头结点的单链表，判定该表为空表的条件是( )。

A．head=NULL B．head->next=NULL

C．head->next==head D．head!=NULL

5．已知串s=“ABCDEFGH'’，则s的所有不同子串的个数为( )。

A．8 B．9 C．36 D．37

三、填空题(15分)

1．假设一个12阶的上三角矩阵A按行优先顺序压缩存储在一维数组B中，则非零元素a8，9，在B中的存储位置k= 。

2．在拓扑排序中，拓扑序列的第一个顶点必定是 的顶点。

3．由六个分别带权值为5，12，9，30，7，16的叶子结点构造一棵哈夫曼(Huffinan)树，该树的结点个数为 ，树的带权路径长度为 。

4．在 的情况下，链队列的出队操作需要修改尾指针。

5．对表长为n的顺序表进行分块查找，若以顺序查找确定块，且每块长度为s，则在等概率查找的情况下，查找成功时的平均查找长度为 。

四、简答题(50分)

1．(6分)已知广义表L：((a，b)，(c，(d，(e)))，f)。

(1)试利用广义表取表头head(L)和取表尾tail(L)的基本运算，将原子c从下列广义表中分解出来，请写出运算表达式。

(2)请给出下列表达式的运算结果：

①hem(tail(head(tail(L)))) ②tail(tail(head(L)))

2．(10分)已知一个图G=(V，E)，其中：

V={a，b，c，d，e，f}；

E={<a，b>，<a，d>，<a，e>，<d，e>，<e，b>，<c，b>，<c，e>，<c，f>，<f，e>}

(1)请画出该图，并写出邻接矩阵。

(2)根据邻接矩阵，分别同从顶点a出发的深度优先和广度优先遍历序列。

(3)画出由此得到的深度优先和广度优先生成树(或森林)。

3．(6分)已知一个栈s的输入序列为abcd，下面两个序列能否通过栈的PUSH和POP操作输出；如果能，请写出操作序列；如果不能，请说明原因。

①dbca ②cbda

4．(5分)设模式串t=”abaabacababa”，试求出t的next和nextval函数值。

5．(7分)已知一组关键字K={20，15，4，18，9，6，25，12，3，22}，请判断此序列是否为堆?如果不是，请调整为堆。

6．(8分)对于表达式(a-b+c)\*d/(e+f)

(1)画出它的中序二叉树，并标出该二叉树的前序线索；

(2)给出它的前缀表达式和后缀表达式。

7．(8分)设散列长度为9，散列函数为H(k)=k mod 9，给出关键字序列：23，45，14，17，9，29，37，18，25，41，33，采用链地址法解决冲突。

(1)请画出散列表；

(2)求出查找各关键字的比较次数；

(3)计算在等概率情况下，查找成功的平均查找长度。

五、算法填空(10分)

已知图的邻接表存储结构描述如下：

CONST N=图的顶点数：

TYPE arcprt=↑arcnode；

arcnode=RECORD

adjvex：integer；

nextarc：acrptr

END；

vexnode=RECORD

vexdata：char；

firstarc：arcptr

END；

Graph=ARRAY[1．．N]of vexnode；

下面是一个图的深度优先遍历的非递归算法，请在 处填上适当内容，使其成为一个完整算法。

PROCEDURE traver\_dfs(g：graph)；

VAR visited：ARRAY[1．．N] of BOOLEAN；

s：STACK；{s为一个栈}

p：arcptr：

BEGIN

FORi：=1 TO N DO visited[i]：= ①

INISTACK(S)；

FOR i=1 TO N DO

BEGIN

②

BEGIN

visited[i]：=true；

visit(i)； //访问第i个顶点

IF g[i].firstarc≠NIL THEN

PUSH(d，g[i].firstarc)

END

WHILE ③ DO

BECIN

p：POP(s)；

IF p↑.nextarc≠NIL THEN

PUSH(s，p↑.nextarc)；

j：=p↑.adjvex；

IF NOT visited[i] THEN

BEGIN

visited[j]：=true；

visit(j)；

IF e[j].firstarc≠NIL THEN

④

END

END

END

END

中国科学院软件研究所2001年数据结构试题

一、单项选择题(每空2分，共20分)

1．下列函数中渐近时间复杂度最小的是( )。

A．T1(n)=nlog2n+5000n B．T2(n)=n2-8000n

C．T3(n)=nlog221-6000n D．T4(n)=2nlog2n-7000n

2．线性表的静态链表存储结构与顺序存储结构相比优点是( )。

A．所有的操作算法实现简单

B．便于随机存取

C．便于插入和删除

D．便于利用零散的存储器空间

3．设栈的输入序列为1，2，…，n，输出序列为a1，a2，…，an，若存在1≤k≤n使得ak=n，则当k≤i≤n时，ai为( )。

A．n-i+l B．n-(i-k) C．不确定

4．设高度为h的二叉树(根的层数为1)上只有度为0和度为2的节点，则此类二叉树中所包含的节点数至少为( )。

A．2h B．2h-1 C．2h+l D．h+l

5．设指针p指向线索树中的某个结点，则查找\*p在某种次序下的前驱或后继不能获得加速的是( )。

A．前序线索树中查找\*p的前序后继

B．中序线索树中查找\*p的中序后继

C．中序线索树中查找\*p的中序前继

D．后序线索树中查找\*p的后序前继

6．假定有k个关键字互为同义词，若用线性探测法将这k个关键字存入散列表中，至少需要进行( )次探测。

A．k-1 B．k C．k+l D．k(k+1)/2

7．若待排序元素基本有序，则下列排序中平均速度最快的排序是( )；若要求辅助空间为O(1)，则平均速度最快的排序是( )；若要求排序是稳定的，且关键字为实数，则平均速度最快的排序是( )。

A．直接插入排序 B．直接选择排序 C．Shell排序

D．冒泡排序 E．快速排序 F．堆排序

C．归并排序 H．基数排序

8．对于多关键字而言，( )是一种方便而又高效的文件组织文式。

A．顺序文件 B．索引文件 C．散列文件 D．倒排文件

二、问答题(共25分)

1．设A[-2：6；-3：6]是一个用行主序存储的二维数组，已知A[-2，-3]的起始存储位置为loc(-2，-3)=1000，每个数组元素占用4个存储单元，求：(6分)

1)A[4，5]的起始存储位置loc(4，5)；

2)起始存储位置为1184的数组元素的下标。

2．给定二叉树的先序和后序遍历序列，能否重构出该二叉树?若能，试证明之，否则给出反例。(6分)

3．在含有n个关键字的m阶B-树中进行查找，至多读盘多少次?完全平衡的二叉排序树的读盘次数大约比它大多少倍?(设两种树中的每个节点均是一个存储块)。(8分)

4．用向量表示的循环队列的队首和队尾位置分别为1和max\_size，试给出判断队列为空和为满的边界条件。(5分)

三、阅读程序题(共10分)

1．设G是一个有向无环图，试指出下述算法的功能，输出的序列是G的什么序列?(10分)

void Demo (ALGraph G)

{//G是图的逆邻接表，向量outdegree的各分量初值为0。

for(i=0；i<G.NodeNum；i++)

for(p=G.adjlist[i].firstedge；p；p=p->next)

//扫描i的入边表

outdegree[p->adjvex]++； //设p->adjvex=j，则将<i，j>的起点

j的出度加1

initStack(&s)； //设置空栈s

for(i=0；i<G.NodeNum；i++)

if(outdegree[i]==0)

Push(&s，i)； //出度为0的顶点i入栈

while(!StackEmpty(s)) //栈s非空

{ i=Pop(&Q)； //出栈，相当于删去顶点i

prinft(“%c”，G.adjlist[i].vertex)；//输出顶点i

for(p=G.adjlist[i].firstedge；p；p=p->next)

{ //扫描i的入边表

j=p->adjvex； //j是i的入边<j，i>的起点

outdegree[j]--； //j的出度减1，即删去i的入边<j，i>

if(!outdegree[j])

Push(&s’，j)； //若j的出度为0，则令其入栈

}//endfor

}//endwhile

}//endDemo

四、算法题(每题15分，共45分)

1．试设计算法在O(n)时间内将数组A[1．．n]划分为左右两个部分，使得左边的所有元素值均为奇数，右边的所有元素值均为偶数，要求所使用的辅助存储空间大小为O(1)。

2．试写一递归算法，从大到小输出二叉排序中所有的值不小于x的关键字，要求算法的时间为O(h+m)，其中h为树的高度，m为输出的关键字个数。

3．设G是以邻接表表示的无向图，v0是G中的一个顶点，k是一个正的常数。要求写一算法打印出图中所有与v0有简单路径相通，且路径长度小于等于k的所有顶点(不含v0)，路径长度由路径上的边数来定义。

参考答案

第一章

1．(a)是线性结构，对应的数据逻辑结构图见图1-2。

(b)是树形结构，对应的数据逻辑结构图见图1-3。

(c)是有向图，其数据逻辑结构图见图1-4。

2．(a)O(m×n)

(b)s+=b[i][j]的重复执行次数是n(n+1)／2，时间复杂度是O(n2)

(c)O(log2n)

3．D 4．D 5．D

第二章

一、单项选择题

(1) ～(5) DACAA (6)～(10) BAAAD

二、填空题

(1)链域 (2)插入 (3)删除 (4)n-i+1 (5)n-I (6)n／2 (7)(n-1)／2

(8)h=NULL (9)h->next=NULL (10)p->next=h (11) p->next->prior=s

(12)p->next=s (13)p->prior->next (14)p->prior=s (15)p->next->prior=p

(16)p->prior->next=p (17)p->next->prior=p->prior (18)r->next=p

(19)r=p (20)r->next=q->next

三、应用题

1．顺序表：逻辑上相邻的数据元素其物理存储位置必须紧邻。其优点是：节省存储，可以随机存取。

链表：逻辑上相邻的数据元素其物理存储位置不要求紧邻，用指针来描述结点间的逻辑关系，故插入和删除运算比较方便。

2．在带表头结点的单链表中，头指针指向表头结点，不带表头结点的单链表中，头指针指向表中第一个结点(又称首元)，当进行查找运算时，必须从头指针开始去顺次访问表中每一个元素。

头结点是另设一个结点，其指针域指向存放表中第一个元素的结点，设置头结点的好处是：无需对在第一个结点前插入一个结点或删除一个结点时进行特殊处理，也可以对空表和非空表的运算进行统一处理。

四、算法设计题

1. 先判断表满否，若表未满，则从最后一个元素an开始，逐个与x进行比较，若ai>x(1≤i≤n)，则将ai后移一个位置，直到ai≤x，最后把x插入到这个位置，表长+1。算法描述如下：

#define M 100

int insort(Slist \*L，int z)

{int i；

if(L->n>=M)

{printf(“overflow")；

return 0；}

else

for(i=L->n;i>=0;i--)

if(L->data[i]>x

L->data[i+1]=L->data[i];

else break；

L->data[i+1]=x；

L->n++；

return 1；

}

2．逐个查找单链表中的结点x，并计数。

int number(lnode \*h,int x)

{ int n=0；

while(h)

{if(h->data==x)

n++；

h=h->next；

}

return s；

}

3．前插法建立带表头结点的单链表算法中的tag为输人数据结束标志。

Lnode \*createhh(int tag)

{ int x；

Lnode \***p，\***h=(Lnode \*)malloc(sizeof(Lnode))；

h->next=NULL；

printf(“input x:”)；

scanf(“%d”,&x);

while(x!=tag)

{p=(Lnode\*)malloc(sizeof(Lnode));

p->data=x;

p->next=h->next;

h->next=p;

scanf(“%d”,&x);

}

return h;

}

4.先建立一个表头结点，用尾插法建立该单链表。然后将尾结点的指针域值置为表中第一个结点的首地址，最后释放表头结点。算法描述如下：

Lnode \*createht(int tag)

{ int x；

Lnode \***p，\*r,\***h=(Lnode \*)malloc(sizeof(Lnode))；

r=h;

printf(“input x:”)；

scanf(“%d”,&x);

while(x!=tag)

{p=(Lnode\*)malloc(sizeof(Lnode));

p->data=x;

r->next=p;

r=p;

scanf(“%d”,&x);

}

r->next=h->next;

free(h);

return r;

}

5．设p指向待逆置链表中的第一个结点，先将表头结点的链域置空。顺次取出待逆置链表中的第一个结点，用前插法插入到带表头结点的单链表中。

Void reverseh(Lnode \*h)

{ Lnode \*s,\*p=h->next;

h->next=NULL;

while(p)

{s=p;p=p->next;

s->next=h->next;

h->next=s;

}

}

6.逐个检测顺序表中其值在x和y之间的元素，并计数k,再将其值大于y的元素向前移动k个元素。算法描述如下：

void deletexy(Slist \*a,int x,int y)

{ int i,k=0;

for(i=0;i<a->n;i++)

if(a->data[i]>=x&&a->data[i]<=y)

k++;

else

a->data[i-k]=a->data[i];

a->n=a->n-k;}

第三章

一、单项选择题

(1)-(5) CACCB (6)-(7)BC

二、填空题

(1)栈顶 (2)栈底 (3)队尾 (4)队首 (5)O(n) (6)O(1) (7)O(1)

(8)O(1) (9)front=rear (10)(rear+1)％m=front (11)(rear-front+m)％m

(12) 向是的两端

三、应用题

1．栈和队列是操作位置受限的线性表，即对插入和删除运算的位置加以限制。栈是仅允许在表的一端进行插入和删除运算的线性表，因而是后进先出表。而队列是只允许在表的一端插入，另一端进行删除运算的线性表，因而是先进先出表。

2．循环队列的优点是解决了“假上溢”问题。

设用数组A[m]来存放循环队列，设front指向实际队首，rear指向新元素应插入的位置时，front=rear即可能是队空，又可能是队满，为了区分队空和队满可采用下列三种方案：

(1)用“少用一个元素空间”的方案，初态fron=rear=0，判断队空的条件是fron=rear，判断队满的条件是(reaf+1)％m=front。

(2)用“设队空标志”的方法，如设front=-1时为队空标志，初态是front=-1，rear=0。当front=rear时则队满。

(3)用count存放表中元素个数，当count=0时为队空，当count=m时为队满。

3．若进栈序列为a，b，c，d，全部可能的出栈序列是：

abcd，abdc，acbd，acdb，adcb，bacd，badc，bcad，bcda，bdca，cbad，

cbda，cdba，dcba，

不可能的出栈序列为：adbc，bdac，cdab，cadb，cadb，dabc，dacb，

dbac，dbca，dcab。

4．由于队列中操作遵循的原则是先进先出，出队元素的顺序是bdcfea，故元素进队的顺序也是bdcfea，元素进队的顺序与元素出栈的顺序相同，在栈中存取数据遵循的原则是后进先出，要得到出栈序列bdcfea，栈的容量至少是应该存放3个元素的空间。

5. 3 4 25 6 15+-/8\*+

四、算法设计题

1．解本题的基本思想是：先将顺序队列q中所有元素出队，并依次进入顺序栈s中，然后出栈，再依次入队。设队列中的元素为a1,a2,…,an，出队并进栈的序列为a1，a2，…，an，出栈并入队的序列为an，an-1，…，a1，可见顺序队列q中所有元素已逆置了。顺序队列的类型定义为：

#define MAX 100

typedef struct

{int data[MAX]；

int front,rear；

}Squeue；

算法描述如下：

void invert(Squeue \*q)

{int s[MAX],top=0;

while(q->front<q->rear)

s[top++]=q->data[++q->front];

q->front=-1;q->rear=0;

while(top>0)

q->data[q->rear++]=s[--top];

}

2.

(1)递归算法：

int f1(int n)

{ int f;

if(n==0) return (1);

else

return(n\*f1(n/2));

}

(2)将上述递归函数转换成非递归函数时，使用了一个栈，用于存储调用参数和函数值，这里采用一维数组s[n]作为栈，用一个栈顶指针top。算法如下：

#define maxlen 200;

int f2(int n)

{ int s[maxlen];

int top=0,fval=1;

while(n!=0)

{s[top]=n;

n=n/2;

top++;

}

while(top>=0)

{fval=fval\*s[top];

top--;

}

return fval;

}

3.仅设队尾指针的带表头结点的循环链表,找队首结点为rear->next->next。

(1)初始化

Lnode \*int queue()

{Lnode \*rear;

rear=(Lnode\*)malloc(sizeof(Lnode));

rear->next=rear;

return rear;

}

(2)入队算法

Lnode \*enqueue(Lnode \*rear,int x)

{ Lnode \*p=(Lnode\*)malloc(sizeof(Lnode));

p->data=x;

p->next=rear->next;

rear->next=p;

return p;

}

(3)出队算法

int outqueue(Lnode \*rear,int \*x)

{ Londe \*p;

if(rear->next=rear)

{printf(“queue is empty”);

return 0;

}

p=rear->next->next;

\*x=p->data;

rear->next->next=p->next;

free(p);

return 1;

}

第四章

一、单项选择题

(1)一(5)DBDBD (6)-(10)BDBBA (11)-(13)CAA

二、填空题

(1)两个串的长度相等且对应位置上的字符相同。

(2)是由零个字符组成的串 (3)0

(4)”goodmorning” (5)”nin” (6)4 (7)”goto” (8)m

(9)m(n-m+1) (10)Loc(a[0][0])=1000-88=912

(11)n(n+1)/2+1 (12) n(n+1)/2 (13)i(i-1)/2+j-1

(14)(2n-j+2)(j-1)/2+i-j (15)3 (16)4 (17)a(18)((b),((c,(d))))

(19)三元组表 （20）十字链表 （21）子表 (22)子表

三、应用题

1．用联接、取子串、置换运算将串S=”(xyz)+\*”转化为T=”(x+z)\*y”。

C1=substr(S,3,1)=”y” C2=substr(S,6,1)=”+”

C3=substr(S,7,1)=”\*” C4=concat(C3,c11)=”\*y”

T=replace(S,3,1,C2)=replace(S,3,1,substr(S,6,1))=”(x+z)+x”

T=replace(T,6,2,concat(C3,C1))=replace(T,6,2,concat(substr(S,7,1)),substr(S,3,1)))=”(x+z)\*y”。

若只用取子串和联接操作进行的转换过程是：

C1=concat(substr(S,1,2),substr(S,6,1))=”(x+”

C2=concat(substr(S,7,1),substr(S,3,1))=”\*y”

T=concat(concat(concat(substr(S,1,2),substr(S,6,1)),substr(S,4,2)),concat(substr(S,7,1),substr(S,3,1)))

2.串S的长度为n，其中的字符各不相同，所以S=”(x+z)\*y”中含1个字符的子串有n个，2个字符的子串有n-1个……，含n-2个字符的子串有3个，含n-1个字符的子串有2个，共有非平凡子串的个数是n(n+1)/2-1。

3.串T的next和nextval函数值见下表：

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 下标j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 串T | a | b | c | a | A | c | c | B | a | c | a |
| next[j] | 0 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 2 | 1 |
| nextval[j] | 0 | 1 | 1 | 0 | 2 | 2 | 1 | 1 | 0 | 2 | 0 |

4．特殊矩阵是指具有相同值的矩阵元素或零元素的分布具有一定规律，可以将其压缩存储在一维数组中，矩阵元素aij的下标I和j与其在一维数中存放的下标k之间存在一一对应关系，故不会失去随机存取功能。

稀疏矩阵中零元素的分布没有一定规律，可以将非零元素存于三元组表中，非零元素aij在三元组中的存放位置与i、j没有对应关系，故失去随机存取功能。

5．N阶对称矩阵A中，aij=aji，可以仅存放下三角元素（或上三角元素）。设r=max(i,j)，c=min(i,j),

k=r(r-1)/2+c-1;

例如，4阶对称矩阵A，按行优先顺序存于一维数组F[10]中，

0 1 2 3 4 5 6 7 8 9

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a11 | a21 | a22 | a31 | a32 | a33 | a41 | a42 | a43 | a44 |

当i=3，j=2时，k=3\*2/2+2-1=4;

当i=2，j=3时，k=4

(2)当对称矩阵A按列优先顺序压缩存放，若仅存放上三角元素，则有：

k=r(r-1)/2+c-1

例如，4阶对称矩阵A，按列优先顺序存于一维数组F[10]中，

0 1 2 3 4 5 6 7 8 9

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a11 | a12 | a22 | a13 | a23 | a33 | a14 | a24 | a34 | a44 |

当i=1，j=3时，k=3\*2/2+1-1=3;当i=3，j=1时，k=3

第五章

一、单项选择题

(1)-(5)BBCDC (6)-(10)BCABC (11)—(15)DABBD (16)-(19)CCABB

(20)-(24) BBBAC (25)-(27)DBC

二、填空题

（1）有零个或多个 （2）有且仅有一个

（3）根据树的广义表表示，可以画出这棵村，该树的度为4。

（4）树的深度为4

(5)树中叶子结点个数为8

（6）n0=14 (7)n-2m+1 (8)2k-1 (9)2i-1 (10)133 (11)59

(12)25=32 (13)⎡log2(n+1)⎤=⎡log269⎤=7 (14) 25-1+6=37 (15) 19

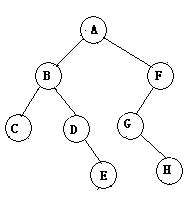
(16)27-1-20=107 (17)右 （18）m+1 (19)n+1 (20) 2m-1

(21)中序 （22）直接前驱结点 （23）直接后继结点

三、应用题

1.具有n个结点的满二叉树中只有度为2和度为0的结点，故n=2n0-1,n0=(n+1)/2。

5．构造一棵二叉树应该先确定根结点，由后序序列最后一个字母A确定根结点，可将前序序列的第一个字母\*改为A。在中序遍历序列中A的左子树中含4个字母，右子树中含3个字母，在后序遍历序列中左子树的后序遍历子序列为\*EDB，可将左子树的后序遍历序列和中序遍历序列互相补足，分别是CEDB和CBDE。由这两个子序列可知左子树的根结点为B，D又为以B为根结点的右子树的根结点，E为D的右孩子，所以在前序序列中A的左子树的前序序列为：BCDE，在根结点A的右子树中，由后序遍历序列可确定其根结点为F，再确定其中序序列为GHF，进一步确定其前序序列为FGH，补足\*处后，三个遍历序列为：

(1)前序遍历序列是：ABCDEFGH

(2)中序遍历序列是：CBDEAGHF

(3)后序遍历序列是：CEDBHGFA

由前序遍历序列和中序遍历序列可构造一棵二叉树如右图。

四、算法设计题

1.求叶子结点个数。

算法描述如下：

int leafnum(Csnode \*t,int n)

{Gsnode \*p;

if(t==NULL) return 0;

p=t->fc;

if(p==NULL)

n++;

while(p)

{n=leaf(p,n);

p=p->ns;

}

return n;

}

2.交换左右孩子。算法如下：

void exchange(Btnode \*t)

{Btnode \*p;

if(t)

{exchange(t->lchild);

exchange(t->rchild);

if(t->lchild||t->rchild)

{p=t->lchild;

t->lchild=t->rchild;

t->rchild=p;}

}

}

3.void printbtree(Btnode \*r)

{if(r)

{printf(“%c”,r->data);

if(r->lchild||r->rchild)

printf(“(”);

printbtree(r->lchild);

if(r->rchild)

printf(“,”);

printbtree(t->rchild);

printf(“)”);

}

}

4. #define MAX 100

int checkt(Btnode \*t)

{Btnode \*s[MAX];

int i,n=0;

for(i=0;i<MAX,i++)

s[i]=NULL;

if(t==NULL)

return 0;

i=0;

S[0]=t;

While(I<=n)

{if(!s[i])

return 0;

if(s[i]->lchild)

{n=2\*i+1;

s[n]=s[i]->lchild;

}

if(s[i]->rchild)

{n=2\*i+2;

s[n]=s[i]->rchild;

}

i++;

}

return 1;

}

5．(1)前序遍历二叉树的非递归算法的基本思想是：从二叉树的根结点开始，沿左支一直走到没有左孩子的结点为止，在走的过程中访问所遇结点，并把非空右孩子进栈。当找到没有左孩子的结点时，从栈顶退出某结点的右孩子，此时该结点的左子树已遍历结束，然后按上述过程遍历该结点的右子树，如此重复，直到二叉树中所有结点都访问完毕为上。算法如下：

void preorder(Btnode \*t)

{ int I=0;

Btnode \*p,\*s[M];

P=t;

Do

{ while(p)

{printf(“%c\t”,p->data);

if(p->rchild!=NULL)

s[i++]=p->rchild;

p=p->lchild;

}

if(i>0)

p=s[--i];

}while(i>0||p!=NULL);

}

(2)后序遍历二叉树的非递归算法的基本思想：采用一个栈保存返回的结点，先遍历根节点的所有结节点并入栈，出栈一个结点，然后遍历该结点的右结点并入栈，再遍历该右结点的所有左结点并入栈，当一个结点的左右子树均访问后再访问该结点，如此这样，直到栈为空为止。其中的难点是如何判断一个结点t的右结点已访问过，为此用p保存已访问过的结点(初值为NULL)，若t->right=p成立（在后序遍历中，t的右节点一定是在t之前访问），说明t的左右子树均已访问，现在应访问t。算法如下：

void postorder(Btree \*t)

{ Btree \*s[maxsize];

Btree \*p;

int flag,top=-1;

Do

{ while(t)

{ top++;

s[top]=t;

t=t->left;

}

p=NULL;

flag=1;

while(top=-1&&flag)

{ t=s[top];

if(t->right==p)

{ printf(“%d””,t->data);

top--;

p=t;

}

else

{ t=t->right;

flag=1;

}

}

}while(top!=-1)

}

6.求二叉树中从根结点到p结点之间路径长度的算法描述如下：

#define MAX 100

void path(Btnode \*t,Btnode \*p)

{ Btnode \*s[MAX],\*q=t;

int b[MAX],top=-1;

do {

while(q)

{s[++top]=q;

b[top]=0;

q=q->lchild;

}

if(top->-1&&b[top]==1)

{q=s[top];

if(q==p)

{printf(“根结点到q结点的路径是：”)；

for(I=0;I<=top;I++)

printf(“%c”,s[I]->data);

return;

}

else top--;

}

if(top>-1)

{ p=s[top]->rchild;

b[top]=1;

}

}while(top>0);

}

7.判断以二叉链表存储的二叉树是否为完全二叉树的算法描述如下：

#define MAX 100

void checkt(BTnode \*t)

{ Btnode \*s[MAX];

int i,n=0;

for(i=0;i<MAX;i++)

s[i]=MAX;

if(t==NULL)

return 0;

s[0]=t;

while(i<=n)

{ if(!s[i])

return 0;

if(s[i]->lchild)

{ n=2\*i+1;

s[n]=s[i]->lchild;

}

if(s[i]->rchild)

{ n=2\*i+2;

s[n]=s[i]->rchild;

}

i++;

}

return 1;

}

第六章

一、单项选择题

(1)-(4)AACA (5)-(9)DBCAD (10)-(11)CB

(12)图G是一个非连通分量，至少有两个连通分量。含8个顶点的完全无向图共需28条边，另外一个顶点构成一个连通分量，所以至少含9个顶点。

二、判断题

1．正确2．错误3．正确 4．错误5．正确6．错误 7．正确 8．正确 9．正确

10．正确(若n个顶点依次首尾相接构成一个环的有向强连通图，至少含n条边，即邻接矩阵中至少有n个小于u的非零元素)。

11．错误 12．正确 13．错误 14．正确 15．正确

16．错误 17．错误 18．错误 19．正确

三、填空题

(1)n(n-1)／2 (2)n(n-1) (3)n-1 (4)e (5)2e (6)出边 (7)人边 (8)n (9)有向图 (10)O(n2) (11)O(n2) (12)O(n+e) (13)O(n+e) (14)Kruscal (15)prim

四、算法题

1．将图的深度优先遍历或广度优先遍历算法稍加改造，另设m保存访问的顶点个数，若已访问顶点个数m小于图中顶点个数n，则图G不连通，否则为连通。现将深度优先遍历的非递归算法改造如下：

采用邻接表表示，类型定义如下：

#define MaxV 20

typedef struct node

{int adjvex;

struct node \*next;

}Enode;

typedef struct

{int data;

Enode \*firste;

}Vnode;

typedef struct

{Vnode adjlist[MaxV];

int n;

int e;

}ALGraph;

int connect(ALGraph \*G)

{int i,j,Visited[MaxV],m=0,top=0;

Enode \***p,\***S[MaxV];

For(i=0;i<n;i++)

Visited[i]=0;

Visited[0]=0;

Printf(“%c”,G->adjlist[0].data);

m++;

S[top++]=p->next;

j=p->adjvex;

if(visited[j]==0)

{visited[j]=1;

printf(“%c”,G->adjlist[j].data);

m++;

s[top++]=G->adjlist[j]->firste;

}

}

}

if(m<G->a)

return(0);

else

return(1);}

1. (1)解：顺序计算每个顶点链表的表中结点个数，就是该顶点的出度.

算法如下：

void outdegree(graph g,int v)

{arcnode \*p;

int n=0;

p=g.adjlist[v].firstarc;

while(p)

{ n++;

p=p->nextarc;}

}

return n;

}

void outds(graph g)

{ int i;

printf(“各顶点出度：、\n”)；

for(i=0;i<g.vexnum;i++)

printf(“顶点%d：%d\n”,i,outdegree(g,i));

}

(2)void maxoutds(graph g)

{ int maxv=0,maxds=0i,x;

for(i=0;i<g.vexnum;i++)

{ x=outdegredd(g,i);

if(x>maxds)

{maxds=x;maxv=1;}

}

printf(“最大出度：顶点%d的出度%d”,maxv,maxds);

}

(3)void zerods(graph g)

{ int i,x;

printf(“出度为0的顶点：\n”);

for(i=0;i<g.vexnum;i++)

{ x=outdegree(g,i);

if(x==0)

printf(“%d”,i);

}

}

(4)void arc(graph g)

{ int i,j;

arcnode \*p;

printf(“输入边：\n”);

scanf(“%d,%d”,i,j);

p=g.adjlist[i].firstarc;

while(p!=NULL&&p->adjvex!=j)

p=p->nextarc;

if(p==NULL)

printf(“不存在！”)；

else

printf(“存在<%d,%d>边：\n”，i,j)；

}

第七章

一、单项选择题

1. 一(5) DDCCD (6)一(9) DCCC

二、填空题

（1）8 （2）15 （3） 4.3 （4）中序 （5）记录的键值 （6）7

（7）63 （8）17 （9）17 （10）O(n) (11)O(log2n) (12)越大

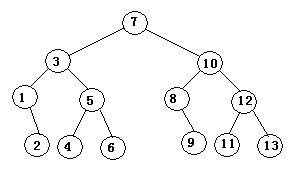
（13）散列函数 （14）处理冲突方法 （15） 装填因子 （16）索引表

（17）该块的起始地址 （18） 100 （19）100 （20）101 （21）⎡log2(n+1)⎤

(22)同义词 （23）n(n+1)/2 (24) 63 (25) m (26)⎡m/2⎤ (27)m-1

(28) ⎡m/2⎤-1 (29)m (30)⎡m/2⎤

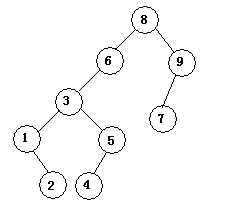
三、应用题

1．判定树见下图。

查找成功的平均查找长度

ASL=（1+2\*2+3\*4+4\*6）/13=41/13

查找失败时与键值最多比较次数为4。

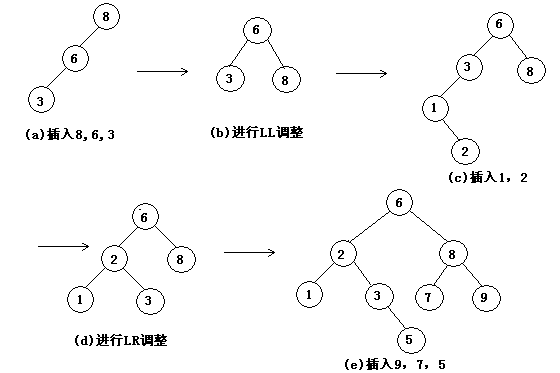


2．二叉排序树见右图。在等概率情况下，

查找成功的平均查找长度ASL=29/9≈3.2

查找失败时，与键值最多比较次数是5。

3.构造一棵平衡二叉排序树

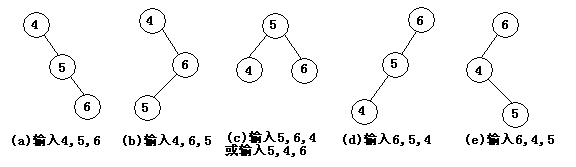
的过程见右图。

等概率情况下，查找成功的平均查找长度

ASL=（1+2\*2+3\*4+4\*2）/9≈2.8

查找失败时，与键值的最多比较次数是4。

4.按不同的输入顺序输入4、5、6，建立相应的不同形态的二叉排序树见下图。



5.键值序列：{25，40，33，47，12，66，72，87，94，22，5，58}

散列地址： 3 7 0 3 1 0 6 10 6 0 5 3

用拉链法处理冲突，构造的散列表如下图：

0

1

2

3

4

5

6

7

8

9

10

11

|  |
| --- |
|  |
|  |
| ^ |
|  |
| ^ |
|  |
|  |
|  |
| ^ |
| ^ |
|  |
| ^ |

|  |  |
| --- | --- |
| 33 |  |

|  |  |
| --- | --- |
| 66 |  |

|  |  |
| --- | --- |
| 22 | ^ |

|  |  |
| --- | --- |
| 12 | ^ |

|  |  |
| --- | --- |
| 47 |  |

|  |  |
| --- | --- |
| 58 | ^ |

|  |  |
| --- | --- |
| 25 |  |

|  |  |
| --- | --- |
| 5 | ^ |

|  |  |
| --- | --- |
| 72 |  |

|  |  |
| --- | --- |
| 87 | ^ |

|  |  |
| --- | --- |
| 40 | ^ |

|  |  |
| --- | --- |
| 94 | ^ |

在等概率的情况下，查找成功的平均查找长度：

ASL=（7+2\*3+3\*2）/12=19/12=1.4

查找失败的平均查找长度：

ASL=（4+2\*1+3\*2）/12=1

6.已知散列表是：

* 1. 1 2 3 4 5 6 7 8 9 10 11 12

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 50 | 21 |  | 57 |  | 45 | 37 |  |

查找key = 21 57 45 50

h0=key%13 = 8 5 6 11

d=key%11+1 = 11 3 2 7

h1=(h0+d)%13 = 6 8 8 5

h2=(h1+d)%13 = 10

查找成功比较次数 2 2 3 2

查找成功的平均查找长度ASL=（1+2+2+2+3）/5=2

四、算法设计

1．按中序遍历二叉排序树可得到一个按键值从小到大排列的有序表，利用这个特点来判别二叉树是否为二叉排序树，算法如下：

#define max 99

#define min 0

int judge(BTnode \*bt)

{ BTnode \*s[max],\*p=bt;

int top=0,preval=min;

do {

while(p)

{s[top++]=p;

p=p->lchild;

}

if(top>0)

{p=s[--top];

if(p->data<preval)

return 0;

preval=p->data;

p=p->rchild;

}

}while(p||top>0)

return 1;

}

第八章

一、单项选择题

1. 一(5) DDCCD (6)一(9) DCCC

二、填空题

（1）8 （2）15 （3） 4.3 （4）中序 （5）记录的键值 （6）7

（7）63 （8）17 （9）17 （10）O(n) (11)O(log2n) (12)越大

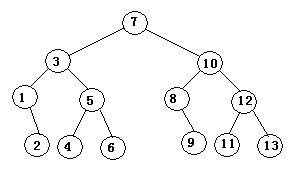
（13）散列函数 （14）处理冲突方法 （15） 装填因子 （16）索引表

（17）该块的起始地址 （18） 100 （19）100 （20）101 （21）⎡log2(n+1)⎤

(22)同义词 （23）n(n+1)/2 (24) 63 (25) m (26)⎡m/2⎤ (27)m-1

(28) ⎡m/2⎤-1 (29)m (30)⎡m/2⎤

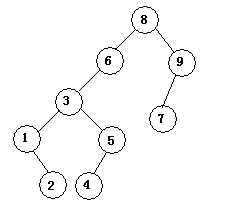
三、应用题

1．判定树见下图。

查找成功的平均查找长度

ASL=（1+2\*2+3\*4+4\*6）/13=41/13

查找失败时与键值最多比较次数为4。

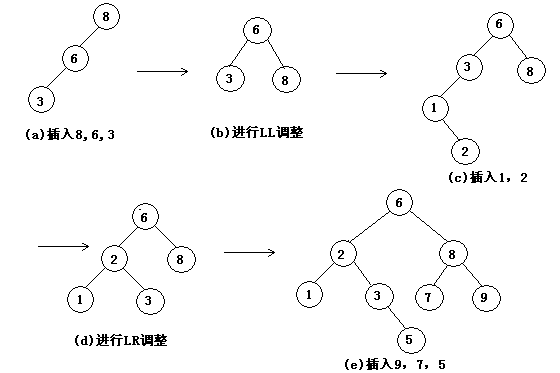


2．二叉排序树见右图。在等概率情况下，

查找成功的平均查找长度ASL=29/9≈3.2

查找失败时，与键值最多比较次数是5。

3.构造一棵平衡二叉排序树

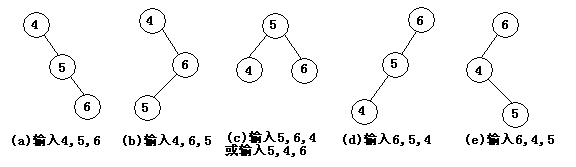
的过程见右图。

等概率情况下，查找成功的平均查找长度

ASL=（1+2\*2+3\*4+4\*2）/9≈2.8

查找失败时，与键值的最多比较次数是4。

4.按不同的输入顺序输入4、5、6，建立相应的不同形态的二叉排序树见下图。



5.键值序列：{25，40，33，47，12，66，72，87，94，22，5，58}

散列地址： 3 7 0 3 1 0 6 10 6 0 5 3

用拉链法处理冲突，构造的散列表如下图：

0

1

2

3

4

5

6

7

8

9

10

11

|  |
| --- |
|  |
|  |
| ^ |
|  |
| ^ |
|  |
|  |
|  |
| ^ |
| ^ |
|  |
| ^ |

|  |  |
| --- | --- |
| 33 |  |

|  |  |
| --- | --- |
| 66 |  |

|  |  |
| --- | --- |
| 22 | ^ |

|  |  |
| --- | --- |
| 12 | ^ |

|  |  |
| --- | --- |
| 47 |  |

|  |  |
| --- | --- |
| 58 | ^ |

|  |  |
| --- | --- |
| 25 |  |

|  |  |
| --- | --- |
| 5 | ^ |

|  |  |
| --- | --- |
| 72 |  |

|  |  |
| --- | --- |
| 87 | ^ |

|  |  |
| --- | --- |
| 40 | ^ |

|  |  |
| --- | --- |
| 94 | ^ |

在等概率的情况下，查找成功的平均查找长度：

ASL=（7+2\*3+3\*2）/12=19/12=1.4

查找失败的平均查找长度：

ASL=（4+2\*1+3\*2）/12=1

6.已知散列表是：

* 1. 1 2 3 4 5 6 7 8 9 10 11 12

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 50 | 21 |  | 57 |  | 45 | 37 |  |

查找key = 21 57 45 50

h0=key%13 = 8 5 6 11

d=key%11+1 = 11 3 2 7

h1=(h0+d)%13 = 6 8 8 5

h2=(h1+d)%13 = 10

查找成功比较次数 2 2 3 2

查找成功的平均查找长度ASL=（1+2+2+2+3）/5=2

四、算法设计

1．按中序遍历二叉排序树可得到一个按键值从小到大排列的有序表，利用这个特点来判别二叉树是否为二叉排序树，算法如下：

#define max 99

#define min 0

int judge(BTnode \*bt)

{ BTnode \*s[max],\*p=bt;

int top=0,preval=min;

do {

while(p)

{s[top++]=p;

p=p->lchild;

}

if(top>0)

{p=s[--top];

if(p->data<preval)

return 0;

preval=p->data;

p=p->rchild;

}

}while(p||top>0)

return 1;

}

第九章

一、单项选择题

(1)-(5) ABABC (6)-(10) ADDAC (11)-(15) BBADB

(16)-(20) DBCAB (21)-(25) ADCBA

二、填空题

（1）插人排序 (2)交换排序 (3)选择排序 (4)归并排序 (5)基数排序

(6)O(nlog2n) (7)O(n2) (8)O(nlog2n) (9)快速 (10)堆 (11)归并

(12)快速 (13)归并 (14)堆 (15)3 (16)选择 (17)O(n) (18)O(nlog2n) (19)O(nlog2n) (20) n2  (22) 5 (23)4 (24) 8

三、应用题

1.解 不稳定的排序方法有快速排序、直接选择排序、堆排序，分别举例说明如下：

（1）快速排序

给定键值序列{9，6，6}进行快速排序。

排序结果为{6，6，9}。

（2）堆排序

给定键值序列为{9,7,6,6}进行堆排序（小顶堆）的过程见下图。

6

7

6

9

9

7

6

6

(6)

9

7

6

(6)

9

7

(6)

(9)

(7)

(6)

(6)

堆排序的结果为{6，6，7，9}。

（3）直接选择排序，例：给定排序码值序列为{6，6，4}，直接选择排序后结果为{4，6，6}。

（4）希尔排序，例给定排序码值序列为{4，2，2，5}，设d1=2,d2=1时，排序结果为{2，2，4，5}。

2.解： 冒泡排序：

第一趟：17，18，40，7，32，60，65，73，85

第二趟：17，18，7，32，40，60，65，73

第三趟：17，7，18，32，40，60，65

第四趟：7，17，18，32，40，60

第五趟：7，17，18，32，40（没有发生交换，结束排序）

3.解：希尔排序：

原序列： 10 18 14 13 16 12 11 9 15 8 d1=5

一趟排序 10 11 9 13 8 12 18 14 15 16 d2=2

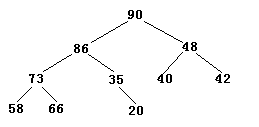
结果

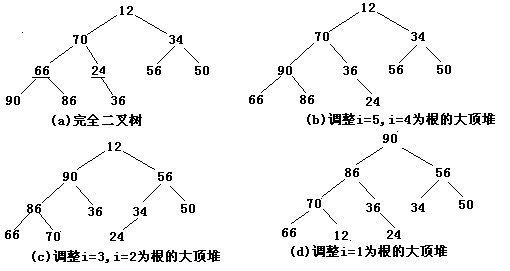
二趟排序 { 8 11 9 12 10 13 15 14 18 16} d3=1

结果

三趟排序结果 {8，9，10，11，12，13，14，15，16，18}

4．初始序列（1）{90，86，48，73，35，40，42，58，66，20}对应的完全二叉树见下图：



它是一个大顶堆。

初始序列（2）{12，70，34，66，24，56，90，86，36}对应的完全二叉树不是堆，将其调整成堆的过程见左图：

四、算法设计题

1. 单链表结点类型定义为：

typedef struct node

{int key;

struct node \*next;

}Lnode;

直接选择排序是每次从n-i+1个结点中选择码值最小者，与第i个结点的码值进行交换，设p指向第i个结点，min指向无序表中码值最小的结点，算法描述如下：

Void selesort1(Lnode \*h)

{Lnode \*p,\*q,\*min;

int x;

p=h->next;

while(p)

{min=p;

q=p->next;

while(q)

{if(q->key<min->key)

m=q;

q=q->next;

}

if(min!=p)

{x=p->key;

p->key=min->key;

min->key=x;

}

p=p->next;

}

}

2.快速排序非递归算法描述如下（其中s为顺序栈）：  
void Quicksort1(RecType t[],int n)

{int top,low=0,high=n,i,j,\*s;

RecType x;

S=(int \*)malloc(sizeof(int)\*2\*n);

Top=0;

Do{

while(low<high)

{i=low;j=high;x=r[i];

do{

while((r[i].key>=x.key)&&(i<j))

j--;

if(i<j){

r[j]=r[i];j--;}

}while(i<j);

r[i]=x;

if((i+1)<high)

{s[++top]=i+1;

s[++top]=high;}

high=i-1;

if(top>0)

{low=s[top--];

high=s[top--];}

}while(top>0||low<high);

}