第一章 绪论

1.1 简述下列术语： 数据、数据项、数据元素、数据逻辑结构、数据存储结构、数据类型、算法。

数据：包含数值型数据与非数值型数据，指有利用价值的信息

数据元素：数据的构成单元

数据项：数据元素内包含的每项数据

1.8 确定下列程序段中标@语句的频度：

(1) i = 1; k=0;

while (i <= n-1) {

@ k+=10\*i; T=n-1

i++;

}

(3) i = 1; k=0;

while (i <= n-1) {

i++;

@ k+=10\*i; T=n-1

}

(5) for (i = 1; i <= n ; i ++) {

for (j = 1; j <= i; j ++)

for (k = 1;k<=j; k++)

@ x+=delta; T=n^4\*1/8 订正：T=(n3+3n2+2n)/6

}

第二章 线性表

2.11 设顺序表Va中的数据元素递增有序，试写一算法，将x插入到顺序表的适当位置上，以保证该表的有序性。

2.14 试写一算法在带头结点的单链表结构上实现线性表操作LENGTH(L)。

2.15

已知指针ha和hb分别指向两个单链表的头结点，并且已知了两个链表的长度分别为m和n，试写一算法将这两个链表连接在一起（即令其中一个表的首元结点连在另一个表的最后一个结点之后），假设指针hc指向连接后的链表的头结点，并要求算法以尽可能短的时间完成连接运算，请分析你的算法的时间复杂度。

2.19

已知线性表中的元素以值递增有序排列，并以单链表作存储结构。试写一高效的算法，删除表中所有值大于mink且小于maxk的元素（若表中存在这样的元素）同时释放被删除结点空间，并分析你的算法的时间复杂度（注意：mink和maxk是给定的两个参变量，它们的值可以和表中的元素相同，也可以不同）。

2.22　试写一算法，对单链表实现就地逆置。

2.24　假设有两个按元素值递增有序排列的线性表A和B，均以单链表作存储结构，请编写算法将A和B表归并成一个按元素值递减有序（即非递增有序，允许表中含有值相同的元素）排列的线性表C，并要求利用表（即A表和B表）的结点空间构造C表。

第三章 栈和队列

3.1　如果进栈的元素序列为1，2，3，4，5，6，能否得到4，3，5，6，1，2和1，3，5，4，2，6的出栈序列？并说明为什么不能得到或如何得到。

3.2　简述栈和线性表的区别。

3.3　写出下列程序段的运行结果（栈中的元素类型是char）：

main(　)

{

SEQSTACK s,\*p;

char x, y;

p = &s;

initstack(p);

x = ′c′; y = ′k′;

push(p,x); push(p,′a′); push(p,y);

x = pop(p);

push(p,′t′); push(p,x);

x = pop(p);

push(p,′s′);

while(!empty(p))

{ y = pop(p);

printf(″%c″,y);}

printf(″%c＼n″,x);

}

3.4 简述以下算法的功能（栈的元素类型为int）

(1) status algo1(Stack S) {

int i,n, A[255];

n=0;

while (!StackEmpty(S)){n++;Pop(S,A[n]);}

for(i=1;i<=n;i++)Push(S,A[i]);

}

(2) status algo2(Stack S,int e){

Stack T;int d;

InitStack(T);

while (!StackEmpty(S)){

Pop(S,d);

if (d!=e) Push(T,d);

}

while (!StackEmpty(T)){

Pop(T,d);

Push(S,d);

}

}

3.7 写出下列式子的逆波兰式

A-B\*C/D+E↑F

3.11　简述栈和队列这两种数据结构的相同点和不同点。

3.12

设Q为顺序队列，InitQueue函数为初始化队列，EnQueue函数为进队，DelQueue函数为出队，QueueEmpty函数为判队列是否为空。，写出以下程序段的输出结果｛队列中的元素类型QElemType为char）。

void main()｛

Queue Q； InitQueue(Q);

Char x＝’e’,y=’c’;

EnQueue(Q,’h’); EnQueue(Q,’r’); EnQueue(Q,y);

DelQueue(Q,x); EnQueue(Q, x);

DelQueue(Q,x); EnQueue(Q,’a’);

while (!QueueEmpty(Q)) {DelQueue(Q,y); printf(“%c”,y);}

printf(“%c”,x);

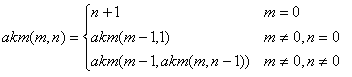
}

3.25 试写出求递归函数F（n）的递归算法，并消除递归

F（n）=n+1 n=0

N\*F(n/2) n>0

3.27 已知Ackerman函数的定义如下：



写出递归算法。

3.28 假设以带头结点的循环链表表示队列，并且只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

4.7　令s=‘aaab’,t=‘abcabaa’，u=’abcaabbabcabaacbacba’。试分别求出它们的next函数值。

第五章

5.10　求下列广义表操作的结果：

(1)GetHead((p,h,w));

(3)GetHead(((a,b),(c,d)));

(5)GetHead(GetTail(((a,b),(c,d))));

5.12　按教科书5.5节中图5.8所示结点结构，画出下列广义表的存储结构图，并求它的深度。

（1）((()),a,((b,c),（），d),(((e))))

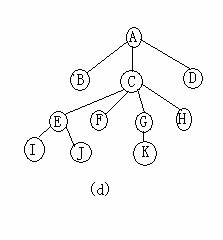
第六章

6.2　一棵度为2的树与一棵二叉树有何区别？

6.3　试分别画出具有3个结点的树和3个结点的二叉树的所有不同形态。

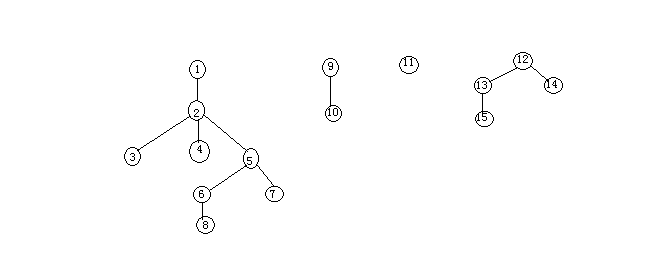
6.5　已知一棵度为k的树中在n1个度为1的结点，n2个度为2的结点，…，nk个度为k的结点，问该树中有多少个叶子结点？

6.19　画出和下列树对应的各个二叉树：



6.20 将下列森林转化为相应的二叉树，并分别按以下说明进行线索化；

（2）中序全线索化；



6.26

假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0.07,0.19,0.02,0.06,0.32,0.03,0.21,0.10。试为这8个字母设计哈夫曼编码,使用0～7的二进制表示形式是另一种编码方案。对于上述实例，比较两种方案的优缺点。

6.27　假设一棵二叉树的先序序列为EBADCFHGIKJ和中序序列为ABCDEFGHIJK请画出该树。

6.28　假设一棵二叉树的中序序列为DCBGEAHFIJK和后序序列为DCEGBFHKJIA。请画出该树。

6.42编写递归算法，计算二叉树中叶子节点的数目。

6.43　编写非递归算法，将二叉树中所有结点的左右子树相互交换。

6.47　编写按层次顺序（同一层自左至右）遍历二叉树的算法。

第七章 图

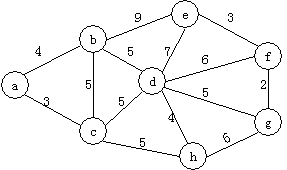
7.3 假定无向图G有6个结点和10条边,这10条边是 (1,2), (1,5), (1,3), (1,6), (2,3), (2,4),

(3,4), (4,5), (4,6),

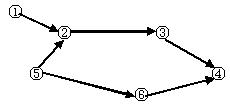
(5,6).画出图G的邻接表和邻接矩阵.并按你所给的邻接表,从结点1出发,分别按深度优先搜索法和广度优先搜索法进行遍历的结点序列。

7.4 试对教课书7.1中的图7.3(a)所示的无向图画出其广度优先生成森林。

7.7 对如下无向带权图：(1) 按普里姆算法求其最小生成树；(2) 按克鲁斯卡尔算法求其最小生成树。



7.9 试列出下图中全部可能的拓扑排序序列。



第九章 查找

9.3　画出长n=10的有序表的进行拆半查找的判定树，并求等概率时查找成功时的平均查找长度。

9.9 已知表(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到一棵初始为空的二叉排序树，画出该二叉排序树。

（2）按表中元素顺序构造一颗平衡二叉树。

9.21　在地址空间为0～12的散列表中,对以下关键字序列构造两个哈希表:(Jan,Feb,Mar,Apr,May,June,July,Aug,Sep,Oct,Nov,Dec)

(1).用线性探测开放定址法处理冲突;(2)用链地址法处理;

并分别求这两个哈希表在等概率情况下查找成功和不成功的平均查找长度.设哈希函数为H(x)= i/2,其中i为关键字中第一个字母在字母表中的序号。

9.31　试写一个判别给定二叉树是否为二叉排序树的算法，设此二叉树以二叉链表存储结构且树中结点的关键字均不同。

9.36　假设二叉树以二叉链表作存储结构，编写在二叉排序中插入一个关键字的算法。

第十章 排序

10.1　对于给定的一组关键字：83, 40, 63,13, 84, 35, 96, 57, 39, 79, 61, 15,

分别画出用直接插入排序、冒泡排序、简单选择排序、快速排序、堆排序、归并排序对上述序列进行操作的各趟结果。

10.2　本章中介绍的各种排序方法中，哪些是稳定排序？哪些是不稳定排序？对不稳定排序方法各举一例说明。

10.23　设立高端监视哨，改写直接插入排序算法。

10.25　试编写教科书10.2.2节中所述链表插入排序的算法。

10.33　试以单链表为存储结构实现简单选择排序的算法。