第 1 章

重点：基本概念和术语、算法效率、算法特性

数据结构：是一门研究非数值计算的程序设计问题中计算机的操作对象以及它们之间的关系和操作等的学科

数据：是指所有能输入到计算机中并被计算机程序处理的符号的总称。

数据元素：是数据的基本单位，在计算机程序中通常作为一个整体进行考虑和处理

数据项：是数据元素的不可分割的最小单位

关键字：能识别一个或多个数据元素的数据项

数据对象：是性质相同的数据元素的集合，是数据的一个子集

数据结构：是相互之间存在一种或多种特定关系的数据元素的集合

逻辑结构：只抽象反映数据元素之间的逻辑关系

- 线性结构—— 线性表、栈、队列、串、数组、广义表（一对一）
- 非线性结构 —— 树和森林和二叉树（一对多）、图（多对多）
- 集合结构 —— 查找表、文件

物理（存储）结构：数据的逻辑结构在计算机存储器中的映象

- 线性结构 —— 顺序、链式、索引、散列（哈希）
- 树形结构 —— 顺序（特定规则下）、链式
- 图形结构 —— 邻接矩阵、邻接表、多重邻接表、十字链表
- 按“关系”的表示方法不同而分：
  - 顺序结构—以数据元素在存储器中的一个固定的相对位置来表示“关系”
  - 链式结构—以指针表示数据元素的“后继”或“前驱”

抽象数据类型：是指一个数学模型以及定义在该模型上的一组操作

构成一个抽象数据类型的三个要素是：数据对象、数据关系、基本操作

算法：是对特定问题求解步骤的一种描述，它是指令的有限序列

算法特性：有穷性、确定性、可行性、输入、输出

设计好的算法目标：有正确性、可读性、健壮性、效率

掌握度量算法效率分析方法：时间、空间复杂度

1．算法的时间复杂度取决于___。

A．问题的规模　　B. 待处理数据的初态　　C. A 和 B

2、(　)数据项是数据处理的最小单位。

第 2 章 线性表

概念

线性表是 n（n>=0）个数据元素的有限序列 。表长度=0，空表。

线性表的特点：存在唯一的一个被称做“第一个”的数据元素

存在唯一的一个被称做“最后一个”的数据元素

除第一个之外，每个元素都只有一个前驱

除最后一个之外，每个元素都只有一个后继

线性表和有序表

—— 不同存储结构的比较

顺序表：逻辑相邻，物理相邻；以“存储位置相邻”表示两个元素间的前驱、后继关系

优点：可以实现随机存取任一元素，O(1)；存储空间使用紧凑

缺点：插入(n/2)和删除((n-1)/2)时需要移动元素；O(n)

需要预分配存储空间; 表容量难扩充

适用于“不常进行修改操作、表中元素相对稳定”的场合。

1．线性表 L=（$a_1,a_2,\ldots,a_n$）用数组表示，假定删除表中任一元素的概率相同，则删除一个元素平均需要移动元素的个数是___。

链表：逻辑相邻，物理不一定相邻（单链表，循环链表，双向链表）

是以“指针”指示后继元素

优点：插入和删除时只需修改指针; O(1)

不需要预分配存储空间;

缺点：只能进行顺序存取；O(n)；指针需额外空间

适用于“修改操作频繁、事先无法估计最大表长”的场合。

有序表：逻辑相邻，物理相邻，且元素值有序排列

3、(　)顺序存储结构中，删除数据元素的操作比较容易。

1、以下数据结构中，从逻辑结构看，（　）和其他数据结构不同。

A．树　　B．顺序表　　C．链队列　　D．循环队列

2、对于链式存储的线性表，查找结点和删除结点的时间复杂度为（　）。

A．O(n) O(n)　B．O(n) O(1)　C．O(1) O(n)　D．O(1) O(1)

4、在存储数据时，通常不仅要存储各数据元素的值，而且还要存储（　）。

A．数据的处理方法　　B．数据元素的类型

C．数据元素之间的关系　　D．数据的存储方法

5、顺序表中删除一个元素，需要平均移动的元素个数为（　）。

A．(n-1)/2　　B．n/2　　C．(n+1)/2　　D．n-1

6、双向链表中有两个指针域，llink 和 rlink，分别指向前驱及后继，设 p 指向链表中的一个结点，q 指向一待插入结点，现要求在 p 前插入 q，则正确的插入为（　）。

A．p->llink=q; q->rlink=p; p->llink->rlink=q; q->llink=p->llink;

B．q->llink=p->llink; p->llink->rlink=q; q->rlink=p; p->llink=q->rlink;

C．q->rlink=p; p->rlink=q; p->llink->rlink=q; q->rlink=p;

D．p->llink->rlink=q; q->rlink=p; q->llink=p->llink; p->llink=q;

7、若某线性表最常用的操作是存取任一指定序号的元素和在最后进行插入和删除运算，则利用（　）存储方式最节省时间。

A．顺序表　　B．双链表

C．带头结点的双循环链表　　D．单循环链表

算法——查找遍历基础上插入、删除或修改

重点：单链表

1、顺序存储：

用一维数组定义一个线性表

```
#define LIST_INIT_SIZE 100
#define LISTINCREAMENT 10
type struct
 {  ElemType *elem; //指向线性表起始地址的指针
```

```
        int  length;      //线性表实际存放数据长度
        int  listsize;    //线性表申请长度
    }SqList
```

遍历

```
SqList L
i=0;  //0 下标开始
While (i<L. length)     for (i=0;i< L. length;i++)
   {                        {
    L.elem[i]                  L.elem[i]
    i++;
   }                        }
```

2、链式存储

用链表定义一个线性表

```
typedef struct Lnode {
      ElemType data;
      struct Lnode *next;
     }Lnode,*LinkList;
```

遍历

```
LinkList L
p=L;  或者 p=L->next;
while (p)
   {
   p->data
   p=p->next;
   }
```

1、已知带头结点的非空链表 L，链表中结点数据域的值无重复。设计一个算法，在不额外申请空间的条件下，将链表 L 中数值最小的结点移动到该链表的前端，作为其首元素。

链表的结点结构如下：

```
typedef struct LNode{
      ElemType        data;
    struct  LNode   *next;
```

}LNode, *LinkList;

1、在一个单链表 L 中，设计算法用指针 P 返回单链表中数据域最大的结点，并删除该结点。

```
LinkedList L_MAXDEL(LinkedList L)
  //L 是单链表的头结点的指针
  {  pre=q=L; p=L->next;  //设 p 指向最大结点，pre 指向最大结点前驱结点，q 遍历
     while(q->next) //查找至最后一个元素结点
       { if (q->next->data > p->data)
           {pre=q; p= q->next;}
             q=q->next;
       }
        pre->next=p->next;
        return p;
  }//算法结束
```

1、已知 L_1、L_2 分别为两循环单链表的头结点指针，m,n 分别为 L_1、L_2 表中数据结点个数。要求设计一算法，用最快速度将两表合并成一个带头结点的循环单链表。

```
LinkedList L_M(LinkedList &L1, LinkedList L2,int m,int n)
```

1．有一个带头结点的单链表，头指针为 head，它的数据域的类型为整型，而且按自小到大的顺序排列，编写一个算法 insertx_list(linklist *head，int x)，在该链表中插入值为 x 的元素，使该链表仍然有序。

1. 假设循环单链表不空，且无表头结点亦无表头指针，指针 P 指向链表中某结点。请设计一个算法，将 P 所指结点的前趋结点变为 P 所指结点的后继结点。

第 3 章 栈和队列

特殊的线性结构：操作受限的线性结构

——栈：先进后出

判断栈满和栈空 StackEmpty(S) ……

——队列：先进先出 InitQueue(Q)……

循环队列，顺序存储 A[0..m-1]的队列满的条件：rear==(rear+1) mod m

队列空的条件: front==rear

队列元素个数：(rear-front+m) mod m

栈和队列的存储：顺序、链式

重点：特性应用

1、有六个元素 1，2，3，4，5，6 的顺序进栈，如果第一个出栈的元素是 4，则第三个出栈的元素不可能是（ ）。

A．3　　B．2　　C．1　　D．6

2、在程序实现递归调用的时候，一般要对临时变量和地址要进行保存，这通常是一个（ ）结构。

A．堆栈　　B．队列　　C．数组　　D．线性表

3、设用链表作为栈的存储结构，则出栈操作（ ）。

A．必须判别栈是否为满　　B．必须判别栈是否为空

C．判别栈元素的类型　　D．对栈不作任何判别

4、循环队列存储在数组 A[0..m]中，则入队时的尾指针操作为（ ）。

rear=(rear+1)mod(m+1)

5、( )图的广度优先遍历要借助于队列来实现。

6、( )树和二叉树的按层遍历要借助于队列来实现。

7、一般情况下，将递归算法转换成等价的非递归算法应该设置（ ）。

A 堆栈　　B 队列　　C 堆栈或队列　　D 数组

8、在解决计算机主机与打印机之间速度不匹配问题时通常设置一个打印数据缓冲区，主机将要输出的数据依次写入该缓冲区，而打印机则从该缓冲区中取走数据打印。该缓冲区应该是一个（ ）结构。

A 堆栈　　B 队列　　C 数组　　D 线性表

第 4 章 串

串和线性表的差异：

数据对象不同（数据元素限定为单个字符）、基本操作集不同（串的整体作为操作对象）、存储结构不同（定长顺序存储、堆分配、块链式）

字符串：是由零个或多个字符组成的有限序列。

一般记为 $S='a_1a_2a_3\ldots a_n'\ (n\geq 0)$

a_i（$1 \le i \le n$）可以是字母、数字或其它字符

串的长度：串中字符的数目 n 称为串的长度

空串：零个字符的串，长度为零，用 Φ 或’’来表示

空格串：由一个或多个空格组成的串。

子串：串中任意个连续的字符组成的子序列

串相等：当且仅当这两个串的值相等

1、(　)两个串相等，当且仅当其长度相等。

2、字符串“ABCDEF”的子串有（ ）个。

22（6+5+4+3+2+1+1（空串））

3、下面关于串的叙述中，（　）是不正确的。

A．串是字符的有限序列

B．空串是由空格构成的串

C．模式匹配是串的一种重要运算

D．串既可以采用顺序存储，也可以采用链式存储

第 5 章

数组——顺序存储结构

二维到一维的不同映象方法：

注意：0 下标开始：

以行序为主序，地址计算 LOC(i,j)=LOC(0,0)+(i*n+j)*L

以列序为主序，地址计算 LOC(i,j)=LOC(0,0)+(j*m+i)*L

注意：1 下标开始：

以行序为主序，地址计算 LOC(i,j)=LOC(1,1)+ ((i-1)*n+j-1)*L

以列序为主序，地址计算 LOC(i,j)=LOC(1,1)+((j-1)*m+i-1)*L

1、数组 A 中，每个元素 A[i,j]的长度为 3 个字节，行下标 i 从 1 到 8，列下标 j 从 1 到 10，从首地址 SA 开始连续存放在存储器内，该数组按行存放时，元素 A[8,5]的起始位置为___。

A SA+141　B SA+144　C SA+222　D SA+225

2．设有二维数组 A[0..9,0..19]，其每个元素占两个字节，第一个元素的存储地址为 100，若按列优先顺序存储，则元素 A[6,6]存储地址为_____。

矩阵的压缩存储

特殊矩阵（a_{11} 开始）

（上）下三角阵 $k=\begin{cases} i(i-1)/2+j-1 & i\geq j \\ \dfrac{n(n+1)}{2} & i<j \end{cases}$ k=0,n(n+1)/2

对称矩阵 $k=\begin{cases} i(i-1)/2+j-1 & i\geq j \\ j(j-1)/2+i-1 & i<j \end{cases}$ k=0,n(n+1)/2-1

对角阵

稀疏矩阵

稀疏因子 $\delta=\dfrac{t}{m\times n}\leq 0.05$

三元组顺序表、行逻辑链接、十字链表（压缩存储优缺点）

1、有一个 8 阶对称阵 A[0..7][0..7]，采用压缩存储方式进行存储 (以行序为主序)，首地址为 100，每个元素所占的单元个数为 3，则 A[6][6]的地址是（ ）。181

0 下标开始，k=(i+1)*i/2+j=7*6/2+6=21+6=27 k=100+27*3=181

1 下标开始，k=i(i-1)/2+j-1

2、将一个 A[1..100，1..100]的三对角矩阵，按行优先存入一维数组 B[1..298]中，A 中元素 $A_{66,65}$（即该元素下标 i=66，j=65）在 B 数组中的位置 K 为（ ）。

A．194 B．195 C．197 D．196

6．设有一个 10 阶的对称矩阵 A，采用压缩存储方式，以行序为主序存储，a_{11} 为第一元素，其存储地址为 1，每个元素占一个地址空间，则 a_{85} 的地址为___。

A. 13 B. 33 C. 18 D. 40

广义表——多层次的线性结构，是线性表的推广

特性：次序性、长度、层次性、深度、递归等

独有的特性：共享

操作函数：GetHead() GetTail()

存储结构的特点：离散式存储

3．当广义表中的每个元素都是原子时，广义表便成了_____。

1、广义表 A((a,b),((c,d),(e)),f)，取出原子 e 的操作是（ ）。HHTHT(A)

2、已知广义表 LS=((a,b,c),(d,e,f))，运用 head 和 tail 函数取出 LS 中原子 e 的运算是( )。

A head(tail(LS)) B tail(head(LS))

C head(tail(head(tail(LS)))) D head(tail(tail(head(LS))))

7．广义表运算式 GetTail((a,b),(c,d))的操作结果是( )。

A. (c,d) B. c,d C. ((c,d)) D. d

第 6 章 二叉树、树和森林

二叉树

存储结构：顺序、链式

顺序存储：适合完全二叉树或满二叉树

顺序存储特点：i 结点左孩子 2i 右孩子 2i+1

链式存储：二叉链表、三叉链表

链式存储特点：二叉链表，空 n+1 个指针，找孩子容易找双亲难；

三叉链表，找孩子找双亲容易，但指针多占空间；

二叉树、完全二叉树、满二叉树

——特点：每个结点至多有二棵子树

子树有左、右之分，且其次序不能任意颠倒

二叉树不是树的特例，有序

——二叉树的性质及其证明方法

性质(1) 第 i 层至多：2^{i-1}

性质(2) k 层至多：2^k-1

1、一棵树高为 K 的完全二叉树至少有（　）个结点。

A．2^k-1　　B．$2^{k-1}-1$　　C．2^{k-1}　　D．2^k

2．一棵树高为 K 的完全二叉树至少有(　)个结点。

A．2^k-1　　B. $2^{k-1}-1$　　C. 2^{k-1}　　D. 2^k

性质(3) 度为 0 和度为 2 结点：n0=n2+1

注意：性质扩展，三叉树，四叉树……

3、设树 T 的度为 3，其中度为 1、2、3 的结点个数分别为 1、2、5，则 T 中的叶子数为（　）。

A．11　　B．12　　C．13　　D．14

N=n0+n1+n2+n3=n0+1+2+5=n0+8

N=B+1，B=1*1+2*2+5*3=20

N0+8=20+1　N0=21-8=13

4、设树 T 的度为 4，其中度为 1、2、3 和 4 的结点个数分别为 4、2、1、1，则 T 中的叶子数为____。

A 5　　B 6　　C 7　　D 8

5．如某二叉树有 20 个叶子结点，有 30 个结点仅有一个孩子，则该二叉树的总结点数为_____。

性质(4) n 个结点的完全二叉树深度：$\lfloor log_2 n \rfloor+1$

性质(5) 完全二叉树结点间：i 结点左孩子为 2i，右孩子为 2i+1

6、一棵完全二叉树有 521 个结点，则其叶子结点个数为_____。

261 , 深度 10, 第 9 层 2^8=256, 521-(2^9-1)=10，即 256-5+2*5=261

7．在完全二叉树中，编号为 i 和 j 的两个结点处于同一层的条件是_____。

⌊log2i⌋=⌊log2j⌋

8．一棵树 T 中，包括一个度为 1 的结点，两个度为 2 的结点，三个度为 3 的结点，四个度为 4 的结点和若干叶子结点，则 T 的叶结点数为_____。

——遍历二叉树

·何谓“遍历”？对结构中的每个元素都访问到，且只被访问一次

·对非线性结构的遍历需要确定一条搜索路径

·对"二叉树"而言，可以有三条搜索路径：

（1）先上后下的按层次遍历；

（2）先左（子树）后右（子树）的遍历；

（3）先右（子树）后左（子树）的遍历。

·先左后右：先序、中序、后序遍历

·重要性质：已知先序和中序，或中序和后序遍历，可以确定唯一的一棵二叉树

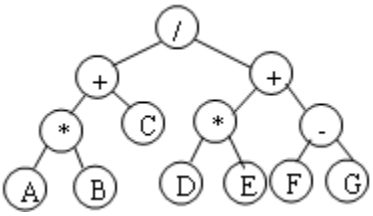
·二叉树可以表示表达式：前缀（先序）、中缀（中序）、后缀（后序），最后进行运算的运算符——树根

9、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

10．设有一表示算术表达式的二叉树，它所表示的算术表达式是(　)。



A. A*B+C/(D*E)+(F-G)　　B. (A*B+C)/(D*E)+(F-G)

C. (A*B+C)/(D*E+(F-G))　　D. A*B+C/D*E+F-G

11．若二叉树采用二叉链表存储结构，要交换其所有分支结点左、右子树的位置，利用(　　)遍历方法最合适。

A．先序　　B．中序　　C．后序　　D．按层次

——算法：递归、非递归

——线索二叉树，利用(n+1)个空闲指针，tag=1 为线索，遍历

掌握“线索化”的过程，画线索二叉树、中序线索二叉树特点

1、(　)中序线索二叉树中，任意一个结点都只有一个前驱和后继。

2、请设计算法求二叉树中元素 X 所在结点的深度。

```
Level=0
int Height(BiTree bt, int level)   //
      {
        if (T)
                {if(T->data == X) return (level)
                height(t->lchild, level+1)；  //左子树深度加 1
                height(t->rchild, level+1)；//右子树深度加 1
                }
        }//结束 height
```

3. 求关键字值为 X 的结点在给定的二叉排序树中所在层次，若无该结点层次为 0。

2．请设计一个算法，要求该算法把二叉树的叶子结点按从左到右的顺序连成一个单链表，表头指针为 head。 二叉树按二叉链表方式存储，链接时用叶子结点的右指针域来存放单链表指针。

树

——三种存储结构：双亲、孩子链表、孩子-兄弟（二叉链表）表示法

——遍历：先根、后根、按层

——树与二叉树的转换

森林

——森林与二叉树的转换、遍历

——遍历：先根、中根

1、若一棵二叉树的先序序列为 ABCHIDEFGKLJ，中序序列为 BHCIAEFDLKGJ。请画出这棵二叉树，并将其转换为对应的森林。

最优二叉树（哈夫曼树）

——构造哈夫曼树及编码（左分支0，右分支1，编码从根结点开始）

——求带权路径长度

7、有 n 个叶子的哈夫曼树的结点总数为（ ）。

A．不确定　　B．2n　　C．2n+1　　D．2n-1

3．假设通信用的报文由 9 个字母 A，B，C，D，E，F，G，H 和 I 构成，它们出现的频率分别是：10，20，5，15，8，2，3，7 和 3。请用这 9 个字母出现的频率作为权值求如下问题：

（1）设计一棵哈夫曼树。

（2）计算其带权路径长度 WPL 值。

（3）写出每个字符的哈夫曼编码。

第 7 章 图

图的定义、术语：无向完全图（n(n-1)/2）边、有向完全图（n(n-1)）弧、稀疏图(e<nlogn)、稠密图、网（带权的图）、连通图、入度、出度、回路

生成树

- 最小生成树：n 个顶点，n-1 条边，代价最小

  选点法（普利姆法）、选边法（克鲁斯卡尔法）

- 拓扑排序：不唯一，可用来判断有向图是否存在环
- AOV 、AOE
- 关键路径：最长路径

深度优先搜索 —— 以结构中的某个数据元素为起始点，首先访问该数据元素，然后依次以它的各个“后继”为起始点进行“深度优先搜索遍历”。

广度优先搜索——以结构中的某个数据元素为起始点，首先访问该数据元素，然后先访问其所有后继；之后其它结点的访问次序由已被访问的结点的访问次序决定：先被访问的结点的后继“优先于”后被访问的结点的后继。遍历可借助队列

图存储结构——邻接表、邻接矩阵、十字链表

1、由 9 个顶点构成的无向非连通图，边的数量最多为（ ）。

28（1 个独立点,1 个完全图）

2．如果含 n 个顶点的图形形成一个环，则它有___棵生成树。

3、在一个 AOE 网中，可以通过（ ）方法判断图中是否有环。

4、( )有向无环图的拓扑序列是唯一的。

5、下列关于 AOE 网的叙述中，不正确的是（ ）。

A．关键活动不按期完成就会影响整个工程的完成时间

B．任何一个关键活动提前完成，那么整个工程将会提前完成

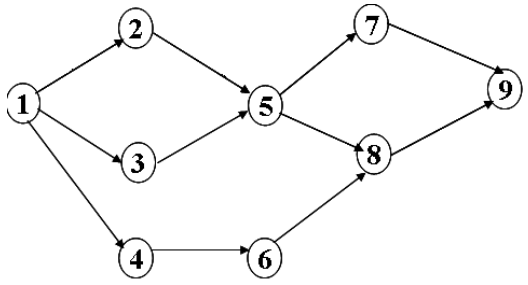
C．所有的关键活动提前完成，那么整个工程将会提前完成

D．某些关键活动提前完成，那么整个工程可能提前完成

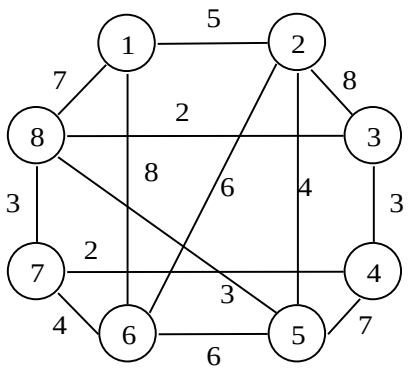
6．已知一无向图 G=（V，E），其中 V={a，b，c，d，e }，E={(a,b)，(a,d)，(a,c)，(d,c)，(b,e)}，现用某一种图遍历方法从顶点 a 开始遍历图，得到的序列为 abecd，则采用的是_____遍历方法。

7．在有向图的邻接矩阵表示中，计算第 i 个顶点入度的方法是_____。第 i 列和

8.拓扑排序序列



9.用克鲁斯卡尔算法构造下图的一棵最小生成树，并给出选边顺序。



1、在图的存储结构里面，图的（　）存储结构相对来说，更适合于稀疏图的存储。

2．用邻接矩阵存储一个图时，在不考虑压缩存储的情况下，所占用的存储空间大小与图中结点个数有关，而与图的边数无关。

（　）

3．下面结构中最适于表示稀疏无向图的是（　）。

A．邻接矩阵　　B．逆邻接表　　C．邻接多重表　　D．十字链表

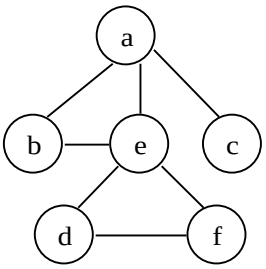
4、在有向图的邻接表存储结构中，结点的个数是图中边个数的（　）倍。

A．1　　B．2　　C．3　　D．4

5、设图如下所示，在下面的 5 个序列中，符合深度优先遍历的序列有（　）。

a e b d f c　　a c f d e b　　a e d f c b　　a e f d c b　　a e f d b c

A．5 个　　B．4 个　　C．3 个　　D．2 个



第 9 章　查找表

·根据查找表所需进行的操作种类和期望达到的 ASL 来选择构造查找表的方法

顺序表、有序表(静态查找树，折半查找、判定树)、索引顺序表 —— 静态查找表、分块查找

二叉排序树、平衡二叉树、平衡因子、构造方法

特性：中序遍历二叉排序树、平衡二叉树，得到的遍历序列及有序序列

7．在任意一棵非空二叉排序树中，删除某结点后又将其插入，则所得二排序叉树与原二排序叉树相同。 （ ）

13．如果按关键码值递增的顺序依次将 n 个关键码值插入到二叉排序树中，则对这样的二叉排序树检索时，平均比较次数为_____。

8．对 n 个元素的表做顺序查找时，若查找每个元素的概率相同，则查找成功的平均查找长度为 。

A．(n+1)/2 B. n/2 C. n D. ((1+n)×n)/2

1、对给定的一组关键字序列(5, 6, 10, 8, 9, 7, 4, 2)，构造一棵平衡二叉树并画图。

1．对字符序列{t,d,e,s,u,g,b,j,a,k,r,i}，构成一棵平衡二叉（排序）树，并为每一次的平衡处理指明旋转类型。（要求画出建树过程）

2、如果按关键码值递减的顺序依次将 n 个关键码值插入到二叉排序树中，则对这样的二叉排序树检索时，平均比较次数为（ ）。

(n+1)/2

3、( )在索引顺序表中，实现分块查找，在等概率查找情况下，其平均查找长度不仅与表中元素个数有关，而且与每块中元素个数有关。

4、有一个长度为 14 的有序表，按折半查找法对该表进行查找，在表内各元素等概率情况下，查找成功所需的平均比较次数为（ ）。

A．37/14 B．39/14 C．43/14 D．45/14

low=1,high=n,mid=⌊(low+high)/2⌋

7. 已知有序表为(12，18，24，35，47，50，62，83，90，115，134)，当用折半查找法查找 100 时，需_____次才能确定不成功。

哈希表 —— 动态查找表也可用于表示静态查找表

各自的特点、操作的实现方法，注意它们之间的相同点和不同点

例如：顺序表的特点是：结构简单，便于插入但不便于删除；平均查找长度较大 ASL=O(n)，哈希表？折半查找（判定树，⌊(low+high)/2⌋）？和动态查找树的区别？

平均查找长度 ASL

·判定树（折半查找）和 ASL 的计算方法

——判定树用于描述查找方法，关键字在判定树上的层次恰为找到它时和给定值进行比较的次数。注意判定树的画法取决于查找方法的本身而不是具体的算法。

1、有一个长度为 15 的有序表，按折半查找法对该表进行查找，在表内各元素等概率情况下，查找成功所需的平均比较次数为（ ）。

A．43/15　　B．45/15　　C．47/15　　D．49/15

·哈希表的特点

是在关键字和记录的存储地址之间建立了一个映象关系，以此减少查找的盲目性，哈希表的最大特点是它的平均查找长度不是表长的函数，因此利用它可以设计出使平均查找长度控制在期望值范围内的查找表。

—— 掌握各种构造哈希表的方法以及处理冲突的方法

1、(　)Hash 表的平均查找长度与处理冲突的方法无关。

2、对以下关键字序列建立哈希表：(12, 32, 16, 19, 24, 27, 56, 67, 35)，哈希表的表长为 12，用二次探测再散列法处理冲突；请给出哈希函数，画出此哈希表，并计算在等概率情况下查找成功的平均查找长度。

表长 m=12，所以 p=11，即 H（key）= key MOD 11

散列地址	0	1	2	3	4	5	6	7	8	9	10	11
关键字	55	12	24	35		16	27		19	67	32	
比较次数	3	1	1	2		1	2		1	5	1	

ASLsucc=(1*5+2*2+3*1+5*1)/9=17/9

3、对以下关键字序列建立哈希表：(29，17，42，19，21，64，23，33)，哈希表的表的装填因子为 0.8，用二次探测再散列法处理冲突；请给出哈希函数，画出此哈希表，并计算在等概率情况下查找成功的平均查找长度。

表长度=8/0.8=10，P 取 7。H（key）= key MOD 7

散列地址	0	1	2	3	4	5	6	7	8	9
关键字	42	29	64	17	33	19	23			21
比较次数	1	1	2	1	3	1	4			3

ASLsucc=16/8

如果表长 m=16，则 p=13

第 10 章 内部排序

·进行排序的目的：得到有序表

直接插入排序、折半插入排序、表插入排序、希尔排序

起泡排序、快速排序

简单选择排序、堆排序

2-路归并排序

基数排序排序

排序和查找相互关联，有时排序的过程也可以看成是一个动态造表的过程，如：插入排序；二叉排序树、平衡二叉排序树

·掌握各种排序方法的特点以便灵活应用

1、给出一组关键字(45, 27, 22, 17, 19, 30, 52, 50)，写出建大顶堆过程（包括每个元素的筛选过程）。

2、给出一组关键字：(25，18，17，30，12，29，16，20，26)分别写出按快速排序方法进行排序时的每一趟变化过程。

3、当初始关键字基本有序的情况下，下列哪种排序方法的时间复杂度受到了影响（ ）。

A．起泡排序　B．简单选择排序　C．归并排序 D．堆排序

注意：选择排序、归并排序、堆排序时间性能不随关键字的分布而改变；直接插入排序、冒泡、快速时间复杂度受到了影响，快速退化 $O(n^2)$,而前两个变为 O(n)

4．分别采用堆排序、快速排序、冒泡排序和归并排序，对初态为有序的表，则最省时间的是_____算法。

冒泡

5．不受待排序初始序列的影响，时间复杂度为 $O(n^2)$的排序算法是_____。

简单选择排序

6、设一组初始记录关键字序列为(50，40，95，20，15，70，60，45)，则以增量 d=4 的一趟希尔排序结束后前 4 条记录关键字为（　）。

A．40，50，20，95　　B．15，40，60，20

C．15，20，40，45　　D．45，40，15，20

7．数据序列（2，1，4，9，8，10，6，20）只能是下列排序算法中的(　)___的两趟排序后的结果。

A. 快速排序　　B. 冒泡排序　C. 选择排序　　D. 插入排序

8．对下列关键字序列用快速排序法进行排序时，速度最快的情形是(　)。

A．{21,25,5,17,9,23,30}　　B．{25,23,30,17,21,5,9}

C．{21,9,17,30,25,23,5}　　D．{5,9,17,21,23,25,30}

9．分别采用堆排序、快速排序、冒泡排序和归并排序，对初态为有序的表，则最费时间的是_____算法。

10．在排序算法的最后一趟开始之前，所有元素都可能不在其最终位置上的排序算法是_____。

11．在初始数据表已经有序时，快速排序算法的时间复杂度为 O(nlog2n)(　)

12．堆排序是稳定的排序方法。　　(　)

13．在任何情况下，归并排序都比直接插入排序快。(　)

·各种排序方法的综合比较

时间性能——平均情况、最坏情况、最好情况

(从关键字的比较和记录的移动两个方面进行分析)

空间性能——需要的辅助空间

排序方法	最好情况	最坏情况	平均情况	辅助存储	稳定排序
直接插入排序	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$	√
折半插入排序	$O(n\log_2 n)$	$O(n^2)$	$O(n^2)$	$O(1)$	√
希尔排序	$O(n^{3/2})$	$O(n^2)$	$O(n^{1.3})$	$O(1)$	×
冒泡排序	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$	√
简单选择排序	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$	×
快速排序	$O(n\log_2 n)$	$O(n^2)$	$O(n\log_2 n)$	$O(\log_2 n)$	×
堆排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(1)$	×
归并排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n)$	√
基数排序	$O(d(n+rd))$	$O(d(n+rd))$	$O(d(n+rd))$	$O(n+rd)$	√