《数据结构》课程复习大纲

1. 单选题（共20小题，每小题1分，共20分）
2. 填空题（共10小题，每空2分，共20分）
3. **应用题（共4小题，每题10分，共40分）**
4. 算法题（共2小题，每小题10分，共20分）

**第一章. 绪论**

**主要内容：**

(1) 数据结构的一些基本概念：数据、数据元素、数据的逻辑结构、物理结构等。

数据元素是数据的基本单位，数据项是数据的最小单位，数据结构是带有结构的各数据元素的集合。

典型例题：

1. 在数据结构中，从逻辑上可以把数据结构分成（ ）。

A．动态结构和静态结构 B．线性结构和非线性结构

C．紧凑结构和非紧凑结构 D．内部结构和外部结构

1. 与数据元素本身的形式、内容、相对位置、个数无关的是数据的（ ）。

A．存储结构 B．存储实现

C．逻辑结构 D．运算实现

(2) 算法的概念和特性。

计算机算法必须具备输入、输出、可行性、确定性和有穷性等5个特性。

(3) 算法时间复杂度和空间复杂度的分析。🞹

1. 如下算法的时间复杂度为（ ）。

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

for(j=0;j<m;j++)

a[i][j]=0;

A．O(1) B．O(n) C．O(m+n) D．O(m\*n)

答案：D

解释：语句a[i][j]=0;的执行次数为m\*n。

1. 如下算法的时间复杂度为（ ）。

x=0;

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

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

x++;

A．O(1) B．O(n) C．O(n2) D．O(log2n)

答案：C

解释：语句x++;的执行次数为n-1+n-2+……＋1= n(n-1)/2。

1. 如下算法的时间复杂度为（ ）。

for(i=0;i<n;i++){x++;s=0;}

1. O(1) B．O(n) C．O(log3n) D．O(n2)
2. 某算法的语句执行频度为（3n+ log2n+ n2+8），其时间复杂度表示( )。 A．O(nlog2n) B．O(n) C．O( log3n) D．O(n2)

常见的算法时间复杂度由小到大依次为：Ο(1)＜Ο(log2n)＜Ο(n)＜Ο(nlog2n)＜Ο(n2)＜Ο(n3)＜ Ο(nk) ＜Ο(2n)

**第2章. 线性表 （重点章节）**

**主要内容：**

1. 非空线性表的特点：除第一个元素无直接前驱，最后一个元素无直接后继外，其他每个数据元素都有且仅有一个直接前驱和直接后继。

例题1：线性表L=(a1，a2,……an)，下列说法正确的是（ ）。

A．每个元素都有一个直接前驱和一个直接后继

B．线性表中至少有一个元素(空表就是0元素)

C．表中诸元素的排列必须是由小到大或由大到小

D．除第一个和最后一个元素外，其余每个元素都有一个且仅有一个直接前驱和直接后继。

1. 顺序表：用一组地址连续的存储单元（数组）依次存放线性表的数据元素。
2. 存储结构特点：逻辑上相邻的数据元素，其物理次序也是相邻的。
3. 地址空间计算：线性表第i个元素的存储位置为LOC(ai)=LOC(a1)+(i-1)\*l，随机存取，时间复杂度均为O(1)。
4. 查找、插入和删除算法
5. 平均查找长度ASL：查找值为e的元素(n+1)/2【(最好情况(查1次)+最差情况)（查n次）/2】;插入新元素n/2【(最好情况(在0个位置插入)+最差情况)（在第n个位置插入）/2】;删除第i个元素(n-1)/2【(最好情况(删除第0个元素)+最差情况)（删除第n-1个元素）/2】;三者时间复杂度均为O(n)。

例题2：顺序表中第6个元素的存储地址是100，每个元素的长度为2，则第1个元素的地址是（ ）。

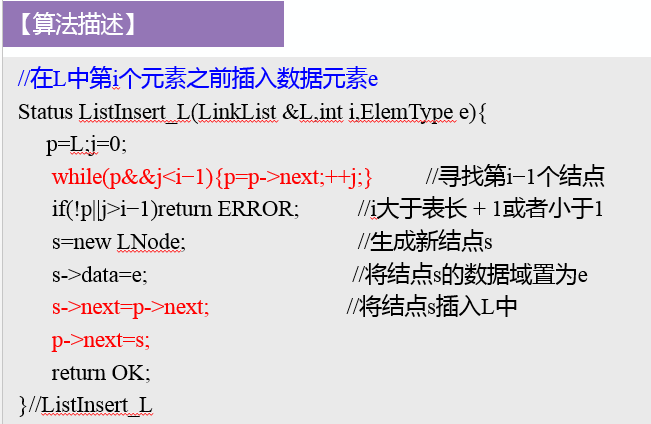
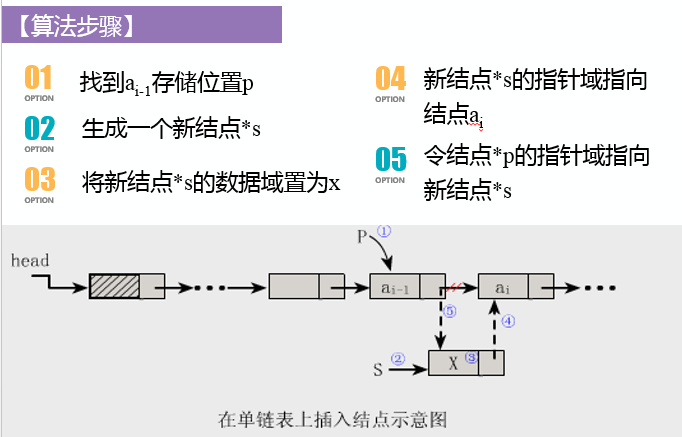
A．88 B．90 C．92 D．96

倒序计算过程：100-（6-1）\*2；

顺序计算过程(设第1个元素的地址为x)：x+(6-1)\*2=100

例题3：向一个有127个元素的顺序表中插入一个新元素并保持原来顺序不变，平均要移动的元素个数为\_\_63.5；【（0+127）/2】在一个长度为n的顺序表中，在第 i（1≤i≤n+1）个元素之前插入一个新元素时，需向后移动\_ n-i+1个元素。【新元素的插入位置为i-1，那么该元素后面的位置为n-(i-1)】

1. 单链表（重中之重）
   1. 存储特点：要取得第i个数据元素必须从头指针出发顺链进行寻找，也称为顺序存取的存取结构。
   2. 头结点、头指针和首元结点的区别
   3. 查找、插入和删除算法



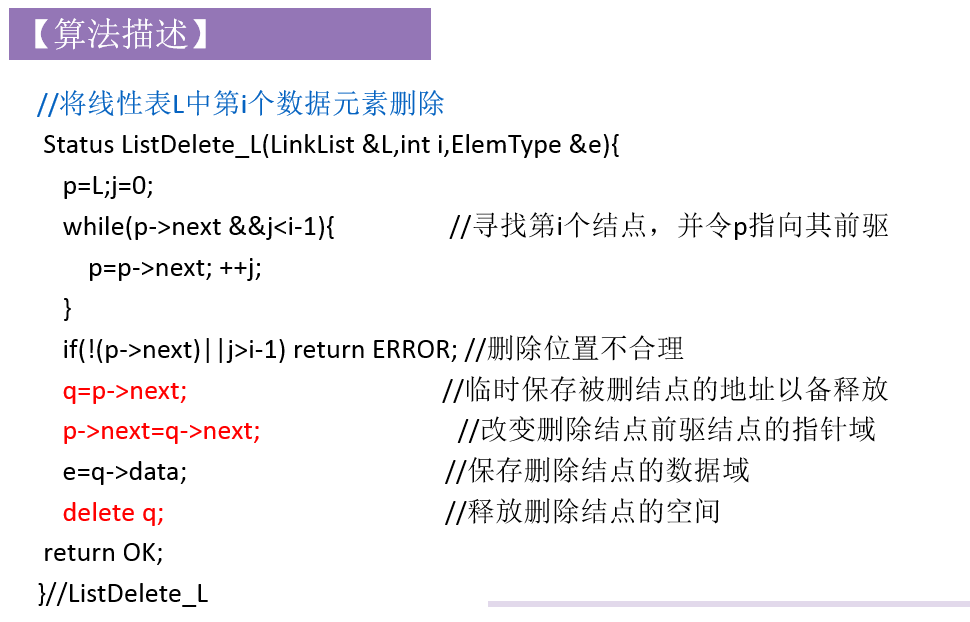
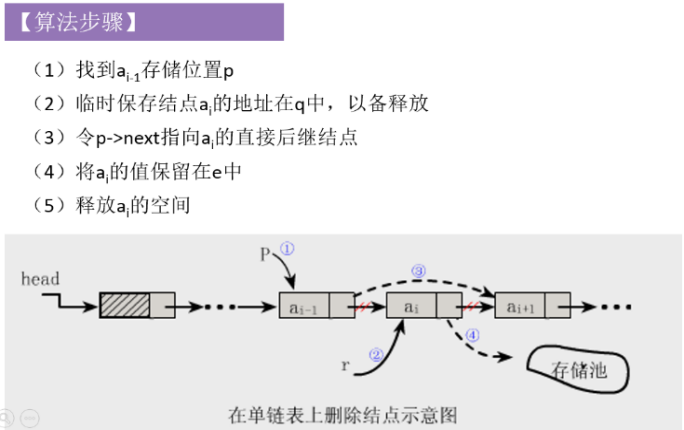
例题4：在单链表中，要将s所指结点插入到p所指结点之后，其语句应为（ ）。

A．s->next=p+1; p->next=s;

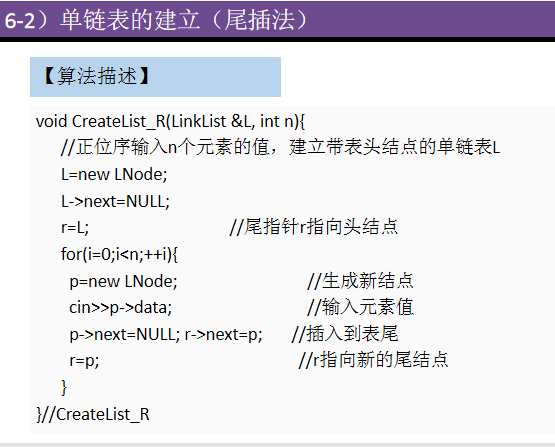
B．(\*p).next=s; (\*s).next=(\*p).next;

C．s->next=p->next; p->next=s->next;

D．s->next=p->next; p->next=s;



（4）单链表的创建（头插法、尾插法）



1. 其它链表
2. 循环单链表

判别当前指针p是否指向表尾结点的终止条件：p!=L或p->next!=L;

1. 双向链表：每个结点包含两个指针域，一个指向直接前趋，一个指向直接后继。
   1. 插入
   2. 删除

例题5：带头结点的循环双向链表L为空表的条件是（ ）。

A. L->next==NULL && L->prior==L

B. L->next==L&&L->prior==NULL

C. L->next==L && L->prior==L

D. L->next== NULL && L->prior== NULL

例题6：在双向链表存储结构中，删除p所指的结点时须修改指针（ ）。

A．p->next->prior=p->prior; p->prior->next=p->next;

B．p->next=p->next->next; p->next->prior=p;

C．p->prior->next=p; p->prior=p->prior->prior;

D．p->prior=p->next->next; p->next=p->prior->prior;

1. 存储密度

存储密度=

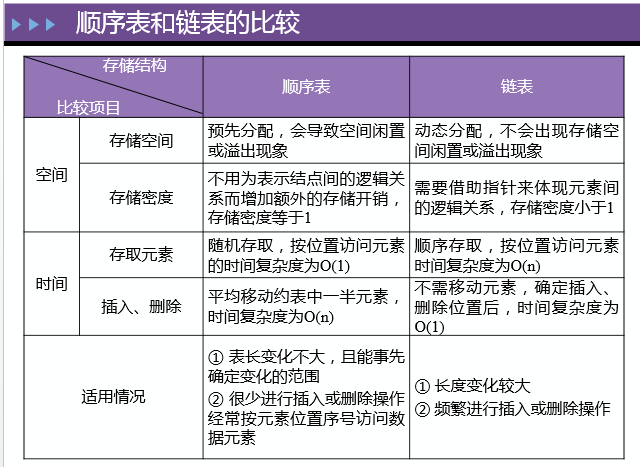
例题7：.单链表的存储密度（ ）。

A．大于1 B．等于1 C．小于1 D．不能确定

(2) 线性表的顺序表示和实现。

(3) 线性表的链式表示和实现。

(4) 线性表的应用，包括无序表和有序表的合并、多项式的加法运算等。



**第三章 栈和队列**

**教学内容：**

(1) 栈的类型定义，栈的顺序存储和链接存储的表示和实现。不带头结点的链栈初始化，就是将栈顶指针top置为空。

(2) 栈的应用举例，如迷宫求解和表达式求值。

(3) 栈与递归的实现，递归程序转换为非递归程序的方法。

例题：

1. 若已知一个栈的入栈序列是1，2，3，…，n，其输出序列为p1，p2，p3，…，pn，若p1=n，则pi为（ ）。

A．i B．n-i C．n-i+1 D．不确定

1. 若让元素1，2，3，4，5依次进栈，则出栈次序不可能出现在（ ）种情况。

A．5，4，3，2，1 B．2，1，5，4，3

C．4，3，1，2，5 D．2，3，5，4，1

1. 若一个栈以向量V[1..n]存储，初始栈顶指针top设为n+1，则元素x进栈的正确操作是( )。

A．top--; V[top]=x; B．V[top]=x; top++;

C．top++; V[top]=x; D．V[top]=x; top--;

1. 在非空的链栈中，栈顶元素出栈，栈顶指针top应修改为：（ ）。

A．top->next=null B．top=0 C．top=top->next D．top=null

1. 栈在（ ）中有所应用。

A．递归调用 B．函数调用

C．表达式求值 D．前三个选项都有

(4) 队列的类型，队列的顺序存储(循环队)和链接存储的表示和实现。

队列的插入和删除运算分别在队列的两端进行，先进队列的元素必定先出队列，所以又把队列称为先进先出（FIFO）表。

例题：

1. 栈和队列的共同点是( )。

A. 都是先进后出 B. 都是先进先出

C. 只允许在端点处插入和删除元素 D. 没有共同点

1. 最大容量为n的循环队列，队尾指针是rear，队头是front，则队空的条件是（　）。

A. rear==front B. (rear+1)%n==front

C．rear+1==front D. (rear-l)%n==front

1. 设采用循环队列存储数据元素，最多时需存储6个元素，问该循环队列数组容量至少需（　 ）个存储空间。

A．6 B．7 C．8 D．5

1. 循环队列存储在数组A[0..m]中，则入队时的操作为（　 ）。

A. rear=rear+1 B. rear=(rear+1)%(m-1)

C. rear=(rear+1)%m D. rear=(rear+1)%(m+1)

**基本要求：**

掌握栈和队列的特点，并能在相应的应用问题中正确选用。熟练掌握栈的顺序栈和链栈的进栈出栈算法，特别应注意栈满和栈空的条件。掌握利用栈实现表达式求值的算法，了解迷宫求解算法。理解递归算法执行过程中栈的状态变化过程，了解将递归程序转换为非递归程序的方法。熟练掌握循环队列和链队列的进队出队算法，特别是循环队列中队头与队尾指针的变化情况。

了解队列的常见应用：打印机的打印数据缓冲区、加油站等等。

**第四章. 串 、数组和广义表**

**教学内容：**

(1) 串的表示和实现，包括顺序存储和链式存储表示。古典的模式匹配算法。

(2) 数组的存储方法。

(3) 特殊矩阵和稀疏矩阵的压缩存储，稀疏矩阵的转置运算。

(4) 广义表的逻辑结构和存储结构。

**基本要求：**

了解串的顺序存储结构和堆存储结构。掌握串的古典的模式匹配算法。掌握数组的地址计算方法。了解稀疏矩阵的两种压缩存储方法的特点和适用范围。了解广义表的结构特点及其存储方法。

典型例题：

1. 串是一种特殊的线性表，其特殊性体现在（ ）。

A．可以顺序存储 B．数据元素是一个字符

C．可以链式存储 D．数据元素可以是多个字符

1. 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．串可采用顺序存储，也可采用链式存储

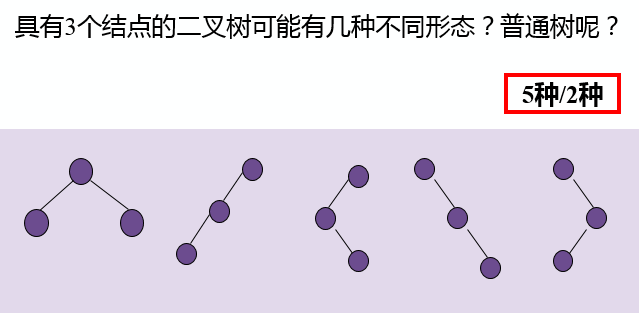
C．模式匹配是串的一种重要运算 D．空串是由空格构成的串

1. 假设以行序为主序存储二维数组A=array[1..100,1..100]，设每个数据元素占2个存储单元，基地址为10，则LOC[5,5]= 818。
2. 广义表((a,b,c,d))的表头是（a,b,c,d）；

设广义表L=((c,d),(f,g))，则L的表尾是 （（f,g）），GetTail(GetHead(L))是 (d) 。

**第五章. 树和二叉树 （重点）**

**教学内容：**

(1) 二叉树的定义和术语，二叉树的性质，特殊的二叉树。

**二叉树的性质**

性质1: 在二叉树的第i层上至多有2i-1个结点

性质2: 深度为k的二叉树至多有2k-1个结点。

性质3: 对于任何一棵二叉树，若2度的结点数有n2个，则叶子数n0必定为n2＋1 （即n0=n2+1）

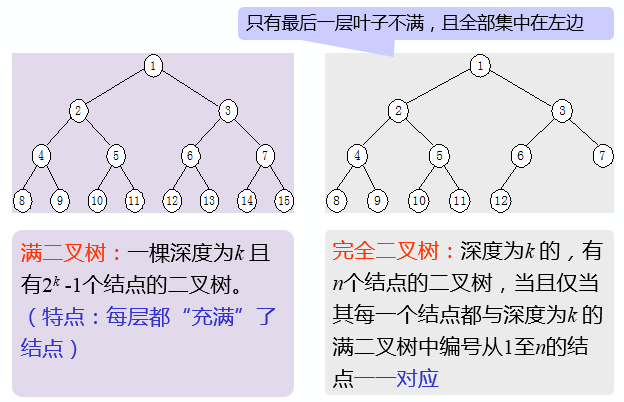
性质4: 具有n个结点的完全二叉树的深度必为⎣ log2n ⎦ ＋1

例题1：设某棵二叉树中有 2000 个结点，则该二叉树的最小高度为 11 。

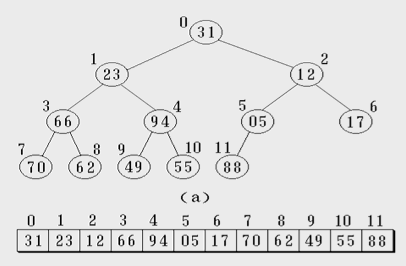
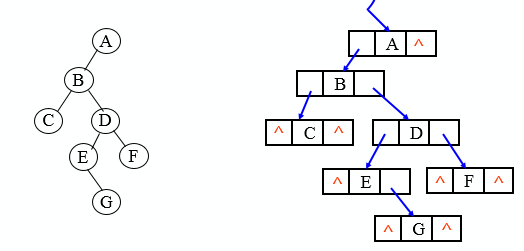
例题2：一个具有1025个结点的二叉树的高h为（ ）。

A．11 B．10 C．11至1025之间 D．10至1024之间

性质5: 对完全二叉树，若从上至下、从左至右编号，则编号为i 的结点，其左孩子编号必为2i，其右孩子编号必为2i＋1；其双亲的编号必为i/2。



(2) 二叉树的存储结构，顺序存储和二叉链表。

二叉树的顺序存储实现：按满二叉树的结点层次编号，依次存放二叉树中的数据元素，适于存满二叉树和完全二叉树。

在n个结点的二叉链表中，必有2n个链域。除根结点外，每个结点有且仅有一个双亲，所以只会有n－1个结点的链域存放指针，指向非空子女结点，有n+1个空指针域。

(3) 二叉树的的前序、中序、后序、层次遍历方法。线索化二叉树。

线索化二叉树：利用二叉链表的空指针，用来存放直接前驱或直接后继，能加快查找结点的前驱或后继的速度。

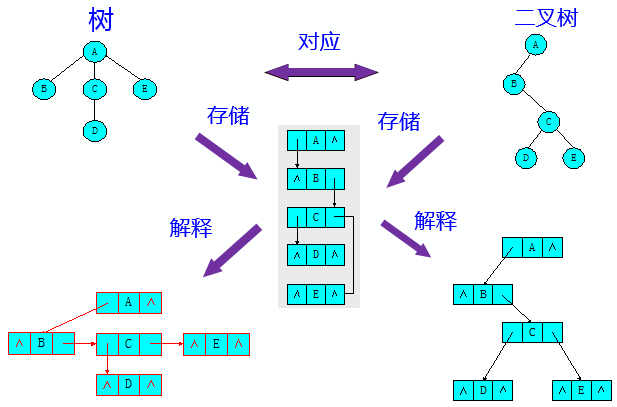
例题：若二叉树采用二叉链表存储结构，要交换其所有分支结点左、右子树的位置，利用 后序 遍历方法最合适。

例题：一棵非空的二叉树的先序遍历序列与后序遍历序列正好相反，则该二叉树一定满足（ ）。

A．所有的结点均无左孩子 B．所有的结点均无右孩子

C．只有一个叶子结点 D．是任意一棵二叉树

例题：已知二叉树先序遍历序列为ABDFCEGH，中序遍历序列为BFDAGEHC，则此二叉树之后序遍历序列为 FDBGHECA 。

(4) 树和森林的定义，树的存储，树、森林与二叉树的转换。

例题：利用二叉链表存储树，则根结点的右指针是（ ）。

A．空 B．指向最右孩子 C．指向最左孩子 D．非空

例题：深度为h的满m叉树的第k层有 mk-1 个结点。(1=<k=<h)

(5) 树的应用，哈夫曼树及哈夫曼编码。

例题：设哈夫曼树中有199个结点，则该哈夫曼树中有（ ）个叶子结点。

A．99 B．100 C．101 D．102

**典型例题：**已知一组权值W=（3,12,7,4,2,8,11）。请完成：

①根据权值画出其哈夫曼树（要求左子树权值小于右子树）；

②计算该哈夫曼树的带权路径长度WPL；

③为各字符设计哈夫曼编码。

【解题思路】

①步骤1：根据给定的n个权值{w1,w2,……wn}，构造n棵只有根结点的二叉树。

步骤2：在森林中选取两棵根结点权值最小的树作左右子树，构造一棵新的二叉树，置新二叉树根结点权值为其左右子树根结点权值之和。

步骤3：在森林中删除这两棵树，同时将新得到的二叉树加入森林中。

步骤4：重复上述两步，直到只含一棵树为止，这棵树即哈夫曼树。



②WPL＝2\*4+3\*4+4\*3+7\*3+8\*3+11\*2+12\*2＝123

再例：已知字符A、B、C、D、E、F权值为27、8、15、15、30、5。

**基本要求：**

了解树和森林的概念，包括树的定义、树的术语。掌握二叉树的概念、性质及二叉树的表示。熟练掌握二叉树的遍历算法，并且能灵活运用遍历算法实现二叉树的其他操作。掌握线索化二叉树的特性及寻找某结点的前驱和后继的方法。了解树的存储、树和森林与二叉树的转换方法。掌握哈夫曼树的实现方法、构造哈夫曼编码的方法及带权路径长度的计算。

**第六章.图**

**教学内容：**

(1) 图的定义和术语。

在一个具有n个顶点的无向完全图中，包含有n\*(n-1)/2条边。

有n个结点的无向图，该图至少应有 n-1 条边才能确保是一个连通图。

G是一个非连通无向图，共有28条边，则该图至少有\_\_\_9\_\_\_个顶点。

例题1：n个顶点的连通图用邻接矩阵表示时，该矩阵至少有（ ）个非零元素。

A．n B．2(n-1) C．n/2 D．n2

例题2：若从无向图的任意一个顶点出发进行一次深度优先搜索可以访问图中所有的顶点，则该图一定是（ ）图。

A．非连通 B．连通 C．强连通 D．有向

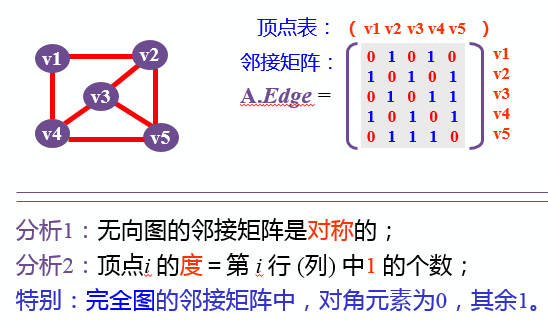
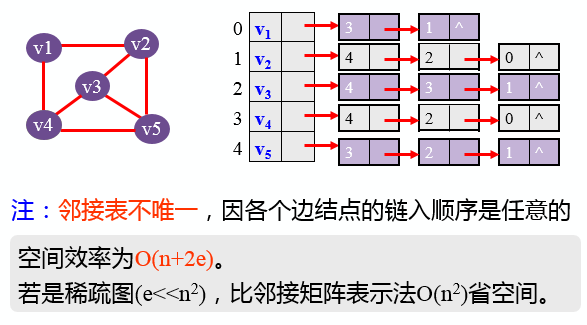
例题3：在一个图中，所有顶点的度数之和等于图的边数的（ ）倍。

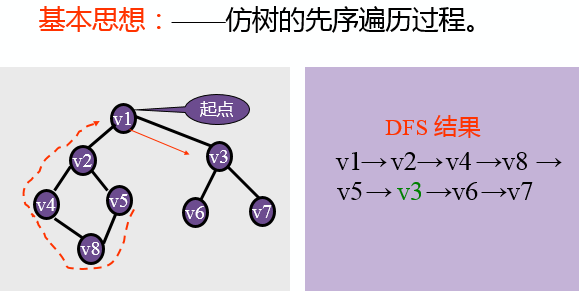
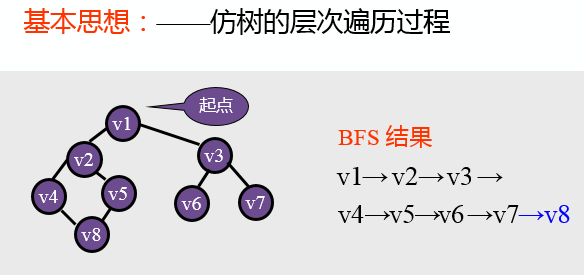
A．2 B．1 C．1/2 D．4

例题4：以下说法正确的是( )。

1. 连通分量是无向图中的极小连通子图
2. 强连通分量是有向图中的极大强连通子图
3. 在一个有向图的拓扑序列中若顶点a在顶点b之前，则图中必有一条弧<a,b>
4. 对有向图G，如果以任一顶点出发进行一次深度优先或广度优先搜索能访问到每个顶点，则该图一定是完全图

(2) 图的存储结构两种存储结构：邻接矩阵和邻接表表示法。



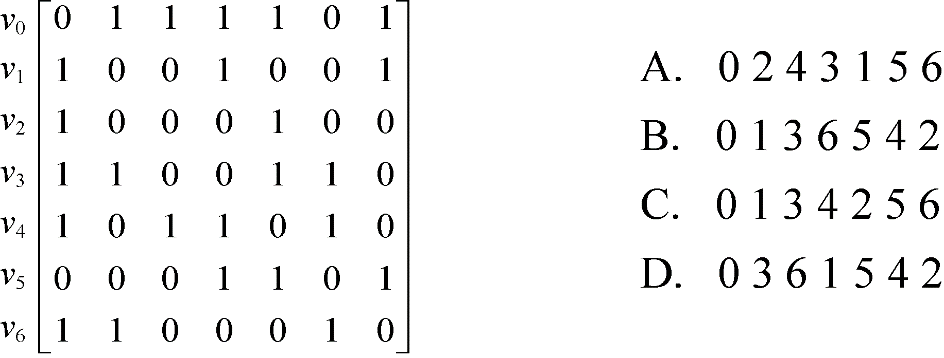
(3) 图的两种遍历策略：深度优先搜索和广度优先搜索。

例题：深度优先遍历类似于二叉树的（ ）。

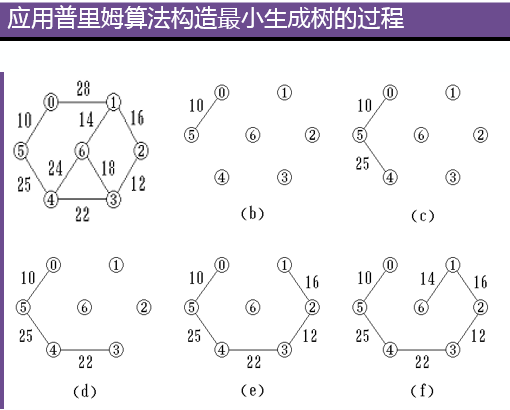
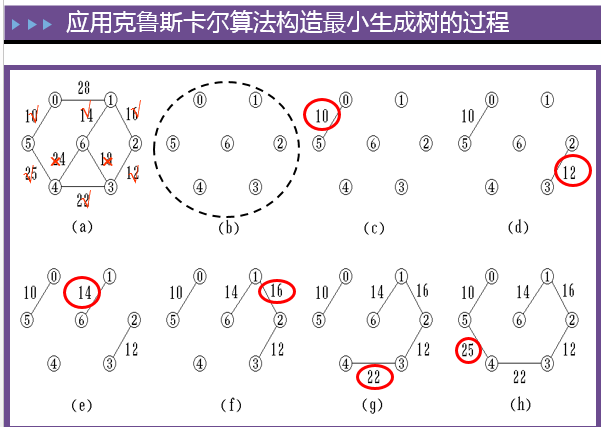
A．先序遍历 B．中序遍历 C．后序遍历 D．层次遍历

例题：图的BFS生成树的树高比DFS生成树的树高（ ）。

A．小 B.相等 C.小或相等 D.大或相等

例题：已知图的邻接矩阵如图所示，则从顶点v0出发按深度优先遍历的结果是0134256。

(4) 构造最小生成树的两种算法：普里姆算法和克鲁斯卡尔算法。



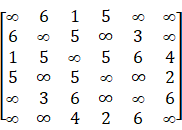


**典型例题：**已知无向连通网图G

①写出其邻接矩阵；

②画出图G从顶点V1开始用普里姆算法构造的最小生成树；

③写出在第②题过程中依次得到的边(连接V1与V2的边用（V1,V2）格式表示)。



从顶点V1开始用普里姆算法构造的过程边序列：(V1，V3),(V3，V6),(V6，V4),(V3，V2),(V2，V5)

用Kruskal算法构造的过程边序列：(V1，V3),(V4，V6),(V2，V5),(V3，V6),(V2，V3)

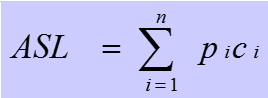
(5) 拓扑排序和关键路径。

(6) 两类求最短路径问题的算法，迪杰斯特拉算法和弗洛伊德算法。

**第7章：查找**

**教学内容：**

(1) 查找的基本概念，平均查找长度。

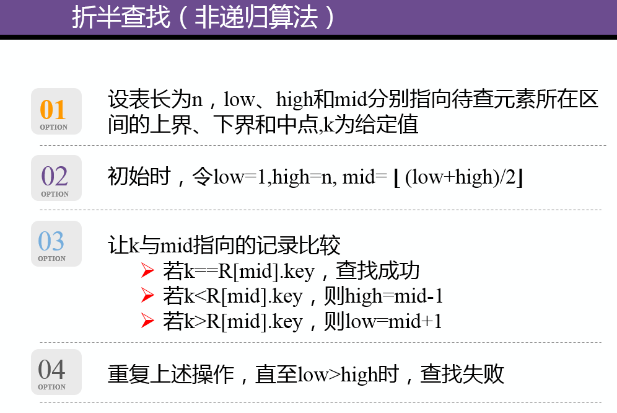
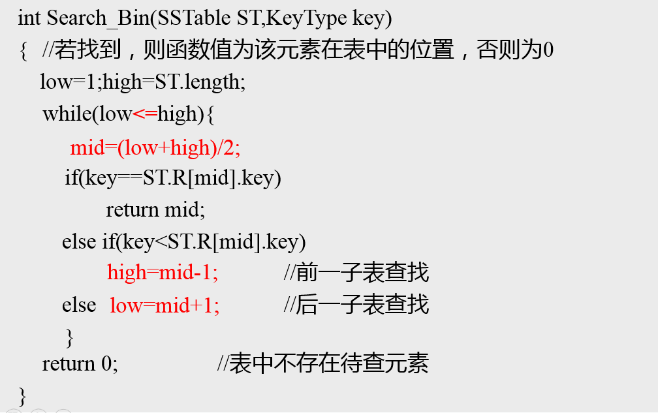
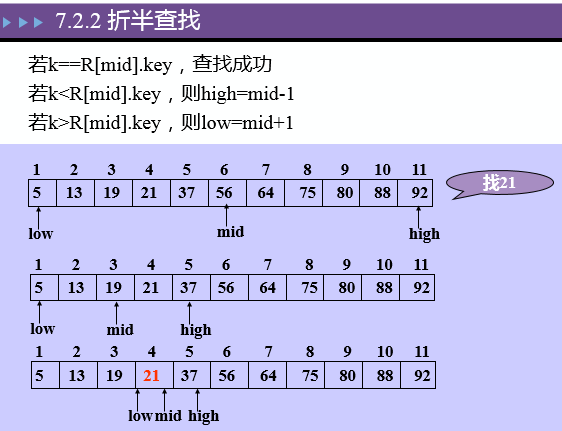
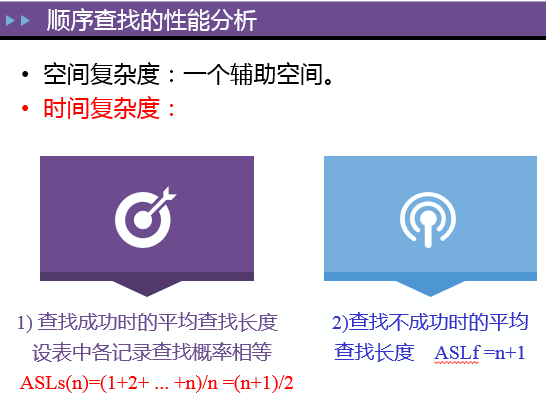
关键字的平均比较次数，也称平均搜索长度*ASL*(*Average Search Length*)

n：记录的个数

pi：查找第i个记录的概率 ( 通常认为pi =1/n )

ci：找到第i个记录所需的比较次数

(2) 基于线性表的查找：顺序查找、**折半查找**。



例题：对22个记录的有序表作折半查找，当查找失败时，至少需要比较 4 次关键字。

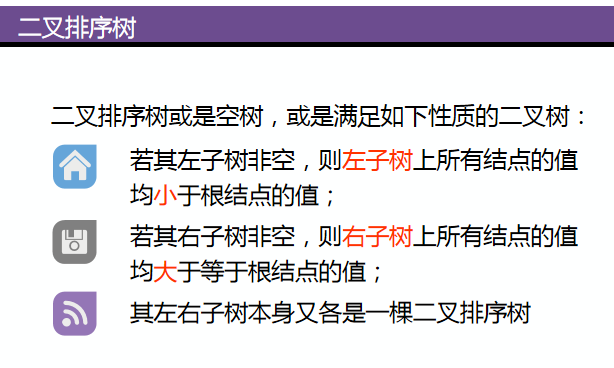
例题：如果要求一个线性表既能较快的查找，又能适应动态变化的要求，最好采用( )查找法。

A．分块查找 B．折半查找

C．顺序查找 D．哈希查找

要求掌握折半查找算法。

（3）二叉排序树



(4)散列表：散列表的基本概念，散列函数的构造方法、处理冲突的方法、散列表的查找与分析。

例题：散列表的平均查找长度( )。

1. 与处理冲突方法有关而与表的长度无关
2. 与处理冲突方法无关而与表的长度有关
3. 与处理冲突方法有关且与表的长度有关
4. 与处理冲突方法无关且与表的长度无关

**典型例题：**设有一组关键字（9，1，23，14，55，20，84，27），采用哈希函数：H（key）=key %7 ，表长为10，用开放地址法的线性探测法处理冲突：

①对这组关键字序列构造哈希表，要求写出计算过程；

②计算查找成功的平均查找长度；

③查找关键字27，需要与哪些元素做对比？

参考：

①对这组关键字序列构造哈希表，要求写出计算过程；

H（9）=2，H（1）=1，H（23）=2，H（14）=0，H（55）=6，H（20）=6，H（84）=0，H（27）=6

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 散列地址 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 关键字 | 14 | 1 | 9 | 23 | 84 |  | 55 | 20 | 27 |  |
| 比较次数 | 1 | 1 | 1 | 2 | 5 |  | 1 | 2 | 3 |  |

②平均查找长度：ASL=（1+1+1+2+5+1+2+3）/8=2

③查找关键字27：需要与55、20、27进行比较，共比较了3次。

如果使用开放地址法的平方探测再散列方法：

, 解决冲突，结果会是怎么样？

**第8章：排序**

**教学内容：**

1. 排序的基本概念

包括正序，逆序，稳定性，排序方法的分类。

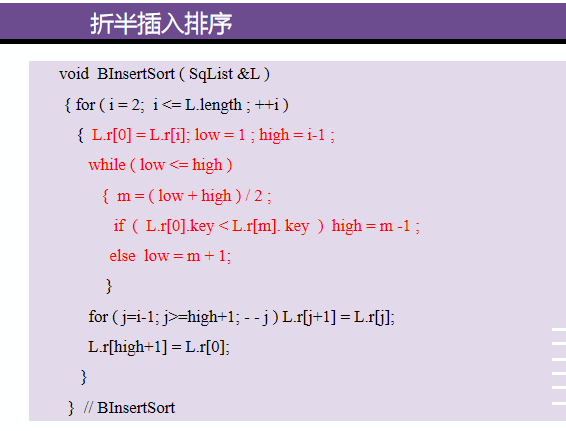
(2) 插入排序：直接插入排序、折半插入排序和希尔排序。

(3) 交换排序：冒泡排序和快速排序。

(4) 选择排序：简单选择排序和堆排序。

(5) 归并排序：2-路归并排序。

(6) 排序算法分析：各种排序算法的比较和移动次数，时间复杂度和空间复杂度的分析。

例题1：从未排序序列中挑选元素，并将其依次放入已排序序列（初始时为空）的一端的方法，称为（ ）。



A．快速排序 B．冒泡排序 C．插入排序 D．选择排序

例题2：在所有排序方法中，关键字比较次数与记录的初始排列无关的是( )。

A.希尔排序 B.冒泡排序 C.插入排序 D.选择排序

例题3：快速排序在下列（ ）情况下最易发挥其长处。

A．被排序的数据中含有多个相同排序码

B．被排序的数据已基本有序

C．被排序的数据完全无序

D．被排序的数据中的最大值和最小值相差悬殊

例题4：下述几种排序方法中，（ ）是稳定的排序方法。

A．希尔排序 B．快速排序 C．归并排序 D．堆排序

典型例题：课后习题P270应用题（1）

设待排序的关键字序列为{12，2，16，30，28，10，16\*，20，6，18}，试分别写出使用以下排序方法，每趟排序结束后关键字序列的状态。

③ 希尔排序（增量选取5，3，1）

④ 冒泡排序

⑤ 快速排序

⑥ 简单选择排序

⑧ 二路归并排序

答案：

③ 希尔排序（增量选取5，3，1）

10 2 16 6 18 12 16\* 20 30 28 （增量选取5）

6 2 12 10 18 16 16\* 20 30 28 （增量选取3）

2 6 10 12 16 16\* 18 20 28 30 （增量选取1）

④ 冒泡排序

2 12 16 28 10 16\* 20 6 18 [30]

2 12 16 10 16\* 20 6 18 [28 30]

2 12 10 16 16\* 6 18 [20 28 30]

2 10 12 16 6 16\* [18 20 28 30]

2 10 12 6 16 [16\* 18 20 28 30]

2 10 6 12 [16 16\* 18 20 28 30]

2 6 10 [12 16 16\* 18 20 28 30]

2 6 10 12 16 16\* 18 20 28 30]

⑤ 快速排序

12 [6 2 10] **12** [28 30 16\* 20 16 18]

6 [2] **6**  [10] 12 [28 30 16\* 20 16 18 ]

28 2 6 10 12 [18 16 16\* 20 ] **28** [30 ]

18 2 6 10 12 [16\* 16]  **18**  [20] 28 30

16\* 2 6 10 12 16\* [16] 18 20 28 30

左子序列递归深度为1，右子序列递归深度为3

⑥ 简单选择排序

2 [12 16 30 28 10 16\* 20 6 18]

2 6 [16 30 28 10 16\* 20 12 18]

2 6 10 [30 28 16 16\* 20 12 18]

2 6 10 12 [28 16 16\* 20 30 18]

2 6 10 12 16 [28 16\* 20 30 18]

2 6 10 12 16 16\* [28 20 30 18]

2 6 10 12 16 16\* 18 [20 30 28]

2 6 10 12 16 16\* 18 20 [28 30]

2 6 10 12 16 16\* 18 20 28 [30]

⑧ 二路归并排序

2 12 16 30 10 28 16 \* 20 6 18

2 12 16 30 10 16\* 20 28 6 18

2 10 12 16 16\* 20 28 30 6 18

2 6 10 12 16 16\* 18 20 28 30

再例：设一组初始记录关键字序列为(49，38，65，97，76，13，27，49\*)，请完成以下排序操作（要求结果非递减排列）：

1. 直接插入排序
2. 简单选择排序
3. 快速排序
4. 冒泡排序