# 第7章：数据结构与算法基础

【考点梳理】

## 考点1、数组与矩阵（★★）

### 【考法分析】

#### 1、本知识点的考查形式主要有：给定一些数组或矩阵，计算对应某个元素的存放位置或位置的表示公式。

### 【要点分析】

#### 1、对于数组或矩阵，存储时注意存储方式是按行存储还是按列存储，二者结果有区别。

#### 2、对于存储位置的计算，可以理解为计算当前位置以要求的存储方式存放时，前面已经存放了多少个元素。

#### 

#### 

### 【备考点拨】

#### 1、对于某些相对繁杂的数组或矩阵，建议可以以前几个特殊的元素带入验证公式，排除错误的选项，直到找出正确选项。

## 考点2、线性表（★★★★★）

### 【考法分析】

#### 1、本知识点的主要考查形式有：对顺序表和链表的一些特点描述判断正误；或对顺序表和链表的一些操作进行对比；对于特殊的线性表队列和栈的一些概念描述判断正误，或二者的出入序列合法性的判断。

### 【要点分析】

#### 1、顺序表和链表的对比：

#### 

#### 2、顺序表：线性表顺序存储，即用一组地址连续的存储单元依次存储线性表中的数据元素，从而使得逻辑上相邻的两个元素，在物理上也相邻。在存储之前，先根据线性表的长度分配连续的物理空间，因此后续不方便扩展。只需要存储数据元素，不需要存储元素的逻辑关系因此存储密度为1。

#### 3、链表：线性表链式存储，即用通过指针链接起来的结点来存储数据元素，存储各数据元素的结点物理上不要求连续，因此后期扩展方便。因为物理上不连续，需要同时存储各元素之间的逻辑关系，存储密度小于1。

#### 4、链表的分类：单链表、双链表、循环链表。

#### 5、特殊的线性表：队列（先进先出）、栈（先进后出）。

#### 6、循环队列：

#### 队空条件：head=tail

#### 队满条件：(tail+1)%size=head

### 【备考点拨】

#### 1、掌握顺序表和链表各自的特点，能够加以区分，并判断相关描述的正确性；

#### 2、了解顺序表和链表一些操作的特殊性和对比；

#### 3、 对于队列和栈，掌握相关的特点和一些特殊的操作、循环队列相关判断公式；

#### 4、掌握队列的入队和出队序列的特点；掌握栈的入栈和出栈序列的特点。

## 考点3、广义表（★★）

### 【考法分析】

#### 1、对于本知识点的主要考查形式有：对相关概念的描述判断正误；给定广义表，指出得到对应结果所需的运算过程。

### 【要点分析】

#### 1、广义表是n个表元素组成的有限序列，是线性表的推广。

#### 2、通常用递归的形式进行定义，记做：LS=(a0, a1,…, an)。

#### 注：其中LS是表名，ai是表元素，它可以是表（称做子表），也可以是数据元素（称为原子）。其中n是广义表的长度（也就是最外层包含的元素个数），n=0的广义表为空表；而递归定义的重数就是广义表的深度，直观地说，就是定义中所含括号的重数（原子的深度为0，空表的深度为1）。

#### 3、基本运算：取表头head(Ls)和取表尾tail(Ls)。

#### 取表头head(Ls)，非空广义表的Ls的第一个元素称为表头，它可以是一个单元素，也可以是一个子表。

#### 取表尾tail(Ls)，非空广义表Ls，除表头元素之外，由其余元素所构成的表称为表尾。非空广义表的表尾必定是一个表。

#### 若有：LS1=（a，（b，c），（d，e））

#### head(LS1)＝a

#### tail(LS1)=(（b，c），（d，e）)

### 【备考点拨】

#### 1、了解广义表相关的一些概念；

#### 2、掌握广义表的相关运算。

## 考点4、树与二叉树（★★★★★）

### 【考法分析】

#### 1、本知识点的主要考查形式有：对数与二叉树的一些概念和特性的描述，判断其正误；对于特殊的二叉树（平衡树、哈弗曼树、满二叉树、排序树等）定义、特性的描述判断正误、或根据题干描述构造特殊的二叉树，找到对应的选项；考查二叉树的遍历结果，或根据遍历序列，找到对应的二叉树。

### 【要点分析】

#### 1、树与二叉树的特性：

#### （1）树的概念：

#### 双亲、孩子和兄弟：结点的子树的根称为该结点的孩子；相应地，该结点称为其子结点的双亲。具有相同双亲的结点互为兄弟

#### 结点的度：一个结点的子树的个数记为该结点的度

#### 叶子节点：也称为终端结点，指度为0的结点

#### 内部结点：指度不为0的结点称为分支节点或非终端节点。除根结点之外，分支结点也称为内部结点

#### 结点的层次：根为第一层，根的孩子为第二层，依次类推，若某节点在第i层，则其孩子结点在第i+1层

#### 树的高度：一颗树的最大层次数记为树的高度（深度）

#### （2）二叉树的重要特性：

#### 1、在二叉树的第i层上最多有2i-1个结点（i≥1）；

#### 2、深度为k的二叉树最多有2k -1个结点（k≥1）；

#### 3、对任何一棵二叉树，如果其叶子结点数为n0，度为2的结点数为n2，则n0=n2+1。

#### 4、如果对一棵有n个结点的完全二叉树的结点按层序编号（从第1层到

#### L log2n ˩+1层，每层从左到右），则对任一结点i（1≤i≤n），有：

#### 如果i=1，则结点i无父结点，是二叉树的根；如果i>1，则父结点是L i/2 ˩ ；

#### 如果2i>n，则结点i为叶子结点，无左子结点；否则，其左子结点是结点2i；

#### 如果2i+1>n，则结点i无右子叶点，否则，其右子结点是结点2i+1。

#### 2、特殊的树

#### 二叉树：二叉树是每个结点最多有两个孩子的有序数，可以为空树，可以只有一个结点。

#### 满二叉树：任何结点，或者是树叶，或者恰有两棵非空子树。

#### 完全二叉树：最多只有最小面的两层结点的度可以小于2，并且最下面一层的结点全都集中在该层左侧的若干位置。

#### 平衡二叉树：树中任一结点的左右子树高度之差不超过1。

#### 查找二叉树：又称之为排序二叉树。任一结点的权值，大于其左孩子结点，小于其右孩子结点。

#### 线索二叉树：在每个结点中增加两个指针域来存放遍历时得到的前驱和后继信息。

#### 最优二叉树：又称为哈弗曼树，它是一类带权路径长度最短的树。

#### 路径是从树中一个结点到另一个结点之间的通路，路径上的分支数目称为路径长度。

#### 树的路径长度是从树根到每一个叶子之间的路径长度之和。结点的带权路径长度为从该结点到树根之间的路径长度与该结点权值的乘积。

#### 树的带权路径长度为树中所有叶子结点的带权路径长度之和。

#### 哈弗曼树的构造：（1）根据给定的权值集合，找出最小的两个权值，构造一棵子树最为其孩子结点，二者权值之和作为根结点；（2）在原集合中删除这两个结点的权值，并引入根节点的权值；（3）重复步骤（1）和步骤（2），直到原权值集合为空。

#### 哈夫曼编码：根据哈夫曼树进行边长编码，编码长度与路径长度相关，左侧分支编码为0（或1），右侧分支编码为1（或0），从根结点到对应叶子结点所有路径分支上的编码记录下来，即为该叶子结点的编码。

#### 3、树的遍历操作：遍历是按某种策略访问树中的每个结点，且仅访问一次的过程。

#### 前序遍历：又称为先序遍历，按根🡪左🡪右的顺序进行遍历。

#### 后序遍历：按左🡪右🡪根的顺序进行遍历。

#### 中序遍历：按左🡪根🡪右的顺序进行遍历。

#### 层次遍历：按层次顺序进行遍历。

### 【备考点拨】

#### 1、掌握树相关的概念和特性；

#### 2、掌握一些特殊的树的定义和特性；

#### 3、掌握哈夫曼树的构造过程，哈夫曼编码的构造。

#### 4、掌握树的遍历，能够根据树的遍历序列反向构造二叉树的过程。

## 考点5、图（★★）

### 【考法分析】

#### 1、本知识点的主要考查形式有：判断给出的关于图的概念、特性的描述是否正确；或根据图的邻接矩阵、邻接表，指出相关图、图的特点、图的遍历；根据图示，指出遍历顺序、拓扑序列。

### 【要点分析】

#### 1、完全图

#### 在无向图中，若每对顶点之间都有一条边相连，则称该图为完全图（complete graph）。

#### 在有向图中，若每对顶点之间都有二条有向边相互连接，则称该图为完全图。

#### 2、图的邻接矩阵表示：用一个n阶方阵R来存放图中各结点的关联信息，其矩阵元素Rij定义为：

#### 

#### 3、图的邻接表表示：首先把每个顶点的邻接顶点用链表示出来，然后用一个一维数组来顺序存储上面每个链表的头指针。

#### 

#### 4、图的遍历：

#### 

#### 5、图的拓扑排序：拓扑排序是将AOV网中的所有顶点排成一个线性序列的过程，并且该序列满足：若在AOV网点中从顶点Vi到Vj有一条路径，则在该线性序列中，顶点Vi必然在顶点Vj之前。

#### 6、最小生成树，是该图的极小联通子图。普里姆算法构造最小生成树过程：

#### （1）去掉所有的连线，将所有顶点放到集合R1中，作为未处理结点集合，另新建集合R2存放已处理结点。

#### （2）选择入度为0的顶点作为起点，放到集合R2中；

#### （3）选择R2到R1最短（代价最小）的路径，同时将对应顶点从R1删除并放到集合R2中。

#### （4）重复步骤3直到R1为空。

#### 

### 【备考点拨】

#### 1、掌握图的相关概念；

#### 2、掌握图的存储；

#### 3、掌握图的遍历；

#### 4、掌握图的拓扑序列求取；

#### 5、了解最小生成树的构造。

## 考点6、排序与查找（★★★★★）

### 【考法分析】

#### 1、本知识点的主要考查形式有：给定情景描述，指出适用的排序方法；指出特定排序方法的时间复杂度、空间复杂度；指出二分查找的比较次数、比较对象、

#### 时间复杂度；指出散列表查找的相关概念描述是否正确。

### 【要点分析】

#### 1、顺序查找的思想：将待查找的关键字为key的元素从头到尾与表中元素进行比较，如果中间存在关键字为key的元素，则返回成功；否则，则查找失败。

#### 2、二分法查找的基本思想是：（设R[low,…,high]是当前的查找区)

#### （1）确定该区间的中点位置：mid=L(low+high)/2˩；

#### （2）将待查的k值与R[mid].key比较，若相等，则查找成功并返回此位置，否则需确定新的查找区间，继续二分查找，具体方法如下。

#### 若R[mid].key＞k，则由表的有序性可知R[mid,…,n].key均大于k，因此若表中存在关键字等于k的结点，则该结点必定是在位置mid左边的子表R[low,…,mid–1]中。因此，新的查找区间是左子表R[low,…,high]，其中high=mid–1。

#### 若R[mid].key<k，则要查找的k必在mid的右子表R[mid+1,…,high]中，即新的查找区间是右子表R[low,…,high]，其中low=mid+1。

#### 若R[mid].key=k，则查找成功，算法结束。

#### （3）下一次查找是针对新的查找区间进行，重复步骤（1）和（2）。

#### （4）在查找过程中，low逐步增加，而high逐步减少。如果high<low，则查找失败，算法结束。

#### 折半查找在查找成功时关键字的比较次数最多为 L log2n +1 ˩ 次。

#### 折半查找的时间复杂度为 O(log2n)次。

#### 【例题】请给出在含有12个元素的有序表｛1，4，10，16，17，18，23，29，33，40，50，51｝中二分查找关键字17的过程。

#### 

#### 比较次数：4次；比较对象a[6]，a[3]，a[4]，a[5]。

#### 3、散列表查找的基本思想是：已知关键字集合U，最大关键字为m，设计一个函数Hash，它以关键字为自变量，关键字的存储地址为因变量，将关键字映射到一个有限的、地址连续的区间T[0..n-1](n<<m)中，这个区间就称为散列表，散列查找中使用的转换函数称为散列函数。

#### 开放定址法是指当构造散列表发生冲突时，使用某种探测手段，产生一个探测的散列地址序列，并且逐个查找此地址中是否存储了数据元素，如果没有，则称该散列地址开放，并将关键字存入，否则冲突，继续查找下一个地址。

#### 例：记录关键码为（3，8，12，17，9），取m=10（存储空间为10），p=5，散列函数h=key%p。

#### 

#### 4、排序分类

#### 

#### 5、直接插入排序：即当插入第i个记录时，R1，R2，…，Ri-1均已排好序，因此，将第i个记录Ri依次与Ri-1，…，R2，R1进行比较，找到合适的位置插入。它简单明了，但速度很慢。

#### 注：对于基本有序的序列，选择直接插入排序方法，时间复杂度近乎线性为：O(n)。

#### 

#### 6、希尔（Shell）排序：先取一个小于n的整数d1作为第一个增量，把文件的全部记录分成d1个组。所有距离为dl的倍数的记录放在同一个组中。先在各组内进行直接插入排序；然后，取第二个增量d2<d1重复上述的分组和排序，直至所取的增量dt=1(dt<dt–l<O<d2<d1)，即所有记录放在同一组中进行直接插入排序为止。该方法实质上是一种分组插入方法。

#### 

#### 7、直接选择排序的过程是，首先在所有记录中选出排序码最小的记录，把它与第1个记录交换，然后在其余的记录内选出排序码最小的记录，与第2个记录交换……依次类推，直到所有记录排完为止。

#### 

#### 8、堆排序

#### （1）堆的定义：设有n个元素的序列{K1，K2，…，Kn}，当且仅当满足下述关系之一时，称之为堆。

#### (i) Ki≤K2 i 且Ki≤K2 i +1；

#### (ii) Ki≥K2 i 且Ki≥K2 i +1 。

#### 其中(i)称为小顶堆，(ii)称为大顶堆。

#### （2）堆排序的基本思想为：先将序列建立堆，然后输出堆顶元素，再将剩下的序列建立堆，然后再输出堆顶元素，依此类推，直到所有元素均输出为止，此时元素输出的序列就是一个有序序列。

#### （3）堆排序的算法步骤如下(以大顶堆为例)：

#### (i)初始时将顺序表R[1..n]中元素建立为一个大顶堆，堆顶位于R[1]，待序区为R[1..n]。

#### (ii)循环执行步骤3～步骤4，共n-1次。

#### (iii)假设为第i 运行，则待序区为R[1..n-i+1]，将堆顶元素R[1]与待序区尾元素R[n-i+1]交换，此时顶点元素被输出，新的待序区为R[1..n-i ]。

#### (iv)待序区对应的堆已经被破坏，将之重新调整为大顶堆。

#### （4）堆排序时间复杂度为：O(nlog2n)，是不稳定的排序。

#### 9、冒泡排序的基本思想是，通过相邻元素之间的比较和交换，将排序码较小的元素逐渐从底部移向顶部。由于整个排序的过程就像水底下的气泡一样逐渐向上冒，因此称为冒泡算法。

#### 

#### 10、快速排序采用的是分治法，其基本思想是将原问题分解成若干个规模更小但结构与原问题相似的子问题。通过递归地解决这些子问题，然后再将这些子问题的解组合成原问题的解。

#### 快速排序通常包括两个步骤：

#### 第一步，在待排序的n个记录中任取一个记录，以该记录的排序码为准，将所有记录都分成两组，第1组都小于该数，第2组都大于该数，如图所示。

#### 第二步，采用相同的方法对左、右两组分别进行排序，直到所有记录都排到相应的位置为止。

#### 

#### 11、归并也称为合并，是将两个或两个以上的有序子表合并成一个新的有序表。若将两个有序表合并成一个有序表，则称为二路合并。合并的过程是：比较A[i]和A[j]的排序码大小，若A[i]的排序码小于等于A[j]的排序码，则将第一个有序表中的元素A[i]复制到R[k]中，并令i和k分别加1；如此循环下去，直到其中一个有序表比较和复制完，然后再将另一个有序表的剩余元素复制到R中。

#### 

#### 12、基数排序是一种借助多关键字排序思想对单逻辑关键字进行排序的方法。基数排序不是基于关键字比较的排序方法，它适合于元素很多而关键字较少的序列。基数的选择和关键字的分解是根据关键字的类型来决定的，例如关键字是十进制数，则按个位、十位来分解。

#### 

### 【备考点拨】

#### 1、掌握顺序查找的相关概念；

#### 2、掌握二分查找的过程；

#### 3、掌握散列表的位置分布和冲突相关的概念；

#### 4、掌握常见排序方法的分类、时间复杂度、空间复杂度、稳定性等；

#### 5、掌握常见排序方法的排序过程（堆排序了解其排序过程）。

## 考点7、时间复杂度与空间复杂度（★★★★★）

### 【考法分析】

#### 1、本知识点的考查形式主要有：根据题干描述的情景，根据排序方法、算法逻辑或相关代码，计算其时间复杂度或空间复杂度；根据递归式，计算其时间复杂度；下午题也会考查根据题干说明和代码，指出时间复杂度。

### 【要点分析】

#### 1、时间复杂度是指程序运行从开始到结束所需要的时间。通常分析时间复杂度的方法是从算法中选取一种对于所研究的问题来说是基本运算的操作，以该操作重复执行的次数作为算法的时间度量。一般来说，算法中原操作重复执行的次数是规模n的某个函数T(n)。由于许多情况下要精确计算T(n)是困难的，因此引入了渐进时间复杂度在数量上估计一个算法的执行时间。其定义如下：

#### 如果存在两个常数c和m，对于所有的n，当n≥m时有f(n)≤cg(n)，则有f(n)=O(g(n))。也就是说，随着n的增大，f(n)渐进地不大于g(n)。例如，一个程序的实际执行时间为T(n)=3n3+2n2+n，则T(n)=O(n3)。

#### 常见的对算法执行所需时间的度量：

#### O(1)<O(log2n)<O(n)<O(nlog2n)<O(n2)<O(n3)<O(2n)

#### 2、常见排序方法的时间复杂度和空间复杂度见考点60介绍；

#### 3、常见算法逻辑的时间复杂度：

#### （1）单个语句，或程序无循环和复杂函数调用：O(1)

#### （2）单层循环：O(n)；双层嵌套循环：O(n2)；三层嵌套循环：O(n3)。

#### （3）树形结构、二分法、构建堆过程：O(log2n)。

#### （4）堆排序、归并排序：O(nlog2n)。

#### （5）所有不同可能的排列组合：O(2n)。

#### 4、主定理求固定形式递归式的时间复杂度：

#### 

### 【备考点拨】

#### 1、掌握常见排序算法的时间复杂度和空间复杂度；

#### 2、掌握常见排序算法、常见算法逻辑（如循环）的时间复杂度；

#### 3、了解主定理求取递归式的时间复杂度。

## 考点7、算法基础及常见算法（★★★★★）

### 【考法分析】

#### 1、本知识点的考查形式主要有：根据题干的情景描述，判断所使用的算法策略；判断算法相关描述是否正确；下午题也会考查根据题干说明和代码，判断算法策略。

### 【要点分析】

#### 算法的特性：

#### （1）有穷性：执行有穷步之后结束。

#### （2）确定性：算法中每一条指令都必须有确切的含义，不能含糊不清。

#### （3）输入（>=0）

#### （3）输出（>=1）

#### （4）有效性(可行性)：算法的每个步骤都能有效执行并能得到确定的结果。例如a=0，b/a就无效

#### 2、分治法

#### （1）特征：把一个问题拆分成多个小规模的相同子问题，一般可用递归解决。

#### （2）经典问题：斐波那契数列、归并排序、快速排序、矩阵乘法、二分搜索、大整数乘法、汉诺塔

#### 3、动态规划法（用于求最优解）

#### （1）特征：划分子问题（最优子结构），并把子问题结果使用数组存储，利用查询子问题结果构造最终问题结果。

#### （2）经典问题：斐波那契数列、矩阵乘法、背包问题、 LCS最长公共子序列

#### 4、回溯法

#### （1）特征：系统的搜索一个问题的所有解或任一解。有试探和回退的过程。

#### （2）经典问题：N皇后问题、迷宫、背包问题

#### 5、贪心法（用于求满意解）

#### （1）特征：局部最优，但整体不见得最优。每步有明确的，既定的策略。

#### （2）经典问题：背包问题（如装箱）、多机调度、找零钱问题

### 【备考点拨】

#### 1、掌握算法的特性、概念；

#### 2、掌握常见算法的特点、适用场景，并能够加以区分。