

# 复习提纲

**题型：**选择 15\*1 分；填空 10\*2 分；综合应用题 5\*10 分；算法填空 1\*6 分；算法设计题 1\*9 分

**试卷自带答题纸**

## 第一部分、绪论（选择+填空）

1. 基本概念和术语：

- ✓ 数据元素：基本单位，在计算机中通常以一个整体出现
- ✓ **数据结构**：相互之间存在一种或多种特定关系的元素的集合（简单解释）。讨论数据结构的目的是为了在计算机中实现对数据结构的操作。（**线性+非线性**）
- ✓ 逻辑结构：元素之间的逻辑关系
- ✓ 存储结构：物理结构，数据结构在计算机中的表示（又称为映像）。依据数据元素之间的关系：顺序映像和非顺序映像→顺序存储结构（相对位置）和链式存储结构（指针）。
- ✓ 抽象数据类型：ADT，是指一个熟悉模型以及定义在该模型上的一组操作。数据对象、数据关系、基本操作

2. 算法：

- ✓ 概念：计算机解决问题的步骤序列。
- ✓ 设计要求/特性：有穷性、正确性（确定性）、可行性（可执行性）、输入、输出
- ✓ 算法的度量：事后统计、事前分析估算（**时间复杂度**：问题的规模、数据的初态有关。  
**空间复杂度**）

## 第二部分、线性结构（所有题类）

1. 线性表

- ✓ 顺序表：元素**紧邻存放**的特点（计算地址、存取数据元素）；插入和删除（操作：数据的移动），最省时间的操作表尾，最费时间的表头，**如果**最常用的操作是查找指定序号的元素和在末尾插入元素则采用顺序表是最合理的结构
- ✓ 单链表：任意的存储单元存储数据元素（存取数据元素：遍历）。结点结构（data+next）；插入和删除（操作：指针不断链），基本的操作也通过遍历完成，**注意是否带头结点**。遍历的条件是**当前指针或者 next 域**是否为空指针。表头进行插入和删除时的最佳结构。
- ✓ 双向链表：插入和删除（**不断链**）P40 页有介绍，但需要灵活应用

2. 栈（顺序栈）

- ✓ 概念：在**端点**处进行插入和删除的线性表。端点：栈底、栈顶
- ✓ 特点：**后进先出**，**注意**：入栈不一定一次性完成。
- ✓ 操作：**入栈和出栈**。栈不满就可以入栈，栈不空就可以出栈，注意：栈指针的状态，书上**栈顶的指针**总是指向栈顶元素的下一个位置。
- ✓ 栈的应用场景：数制的转换、括号的匹配、迷宫求解、图的深度优先搜索、关键路径等
- 3. 队列
- ✓ 特点：**先进先出**的线性表，队头删除、队尾插入（伪溢出）
- ✓ 循环队列：**操作**：队满、队空的条件，队满： $(Q.rear+1)\%MAXSIZE==Q.front$ ；队空： $Q.rear==Q.front$ ；队不满就可以入队，队不空就可以出队，注意：front 和 rear 指针的变化。出队： $Q.front = (Q.front + 1) \% maxsize$ ；入队： $Q.rear = (Q.rear+1) \% maxsize$
- ✓ 链队列：结构定义。**操作**：**入队和出队**。两个指针：队头指针 front，队尾指针 rear 的变化
- ✓ 队列的应用场景：图的广度优先搜索、打印机

### 第三部分、 树形结构（所有题类）

1. 树形结构是一类**重要的非线性结构**。结构中的元素之间存在一个对多个的关系，层次关系
2. 树的定义和术语：结点（包括一个数据元素及若干指向其子树的分支）；结点的度（结点拥有子树数）；树的度（树内各结点的度的最大值）；结点层次（根是第一层，依次类推）；树的深度/高度（树中结点的最大层次）。
3. 二叉树：
  - ✓ 特点：每个结点至多只有两棵子树（二叉树中不存在大于 2 的结点），其子树有左右之分，不能互换（有序树）
  - ✓ 二叉树的性质：5 个性质，重要的是 1~3。1、第 i 层上至多有  $2^{i-1}$  个结点。2、深度为 k 的二叉树至多有  $2^k-1$  个结点。**3、 $n_0=n_2+1$** （证明方法：结点  $n=n_1+n_2+n_0$ ；分支： $n-1=n_1+2n_2$ ）。4、n 个结点的完全二叉树的深度  $\lfloor \log_2 n \rfloor + 1$ 。5、完全二叉树的双亲、孩子结点的推算。
  - ✓ 二叉树的存储结构：顺序存储。链式：**二叉链表**（结点形态，含有 n 个结点的二叉链表中有 n+1 个空链域）；三叉链表；线索链表；双亲链表。
  - ✓ 操作：（二叉树的**遍历**）先序、中序、后序（存储结构：二叉链表；先左后右；递归遍历；中序的非递归遍历，依赖栈结构实现）；创建树。表达式的计算；已知两种序列求另一个序列。
4. 线索二叉树：二叉链表作为存储结构时，只能找到结点的左右孩子信息，不能直接叨叨

结点在任一序列中前驱和后继信息。利用  $n+1$  个空链域存放前驱和后继信息，形成线索链表 P132 结点形态。将二叉树以某种次序遍历使其成为线索二叉树的过程叫**线索化**。

5. 树和森林：树的存储结构（**二叉链表——孩子兄弟表示法**），树/森林 $\leftrightarrow$ 二叉树的转换（二叉树的特点：**形态唯一**，且根结点没有右孩子；转换的方式：存储结构）
6. 哈夫曼树：最优树，只有度为 0 和 2 的结点；**建树、编码、计算压缩比等**。

## 第四部分、图形结构

1. 基本概念和术语：
  - ✓ 图：顶点集和弧集构成的数据结构；弧、弧头、弧尾；
  - ✓ 弧的方向性：图分为有向图（弧）和无向图（边）
  - ✓ 子图、网、
  - ✓ **完全图**： $e=n(n-1)/2$ ；有向完全图： $e=n(n-1)$
  - ✓ 稀疏图和稠密图： $n \log n$
  - ✓ 邻接点：顶点之间存在一条边，称为邻接点
  - ✓ **度**：和顶点关联的边的数目；有向图：入度、出度；
  - ✓ 无向图：**连通图**（任意两个顶点之间有路径相通）和**连通分量**（极大连通子图）
  - ✓ 有向图：强连通图和强连通分量
2. 图的存储结构：
  - ✓ **邻接矩阵**（数组）：有向图和无向图都适用的存储结构（存储稠密图较好）；对一个图（网）会做其邻接矩阵，图的邻接矩阵用 1 和 0 表示网的邻接矩阵用权值和无穷来表示；根据邻接矩阵会求遍历序列，借助于数据结构（栈和队列）；会求度；
  - ✓ **邻接表**、逆邻接表：链式的存储结构；邻接表对有向图和无向图都适用的存储结构（存储稀疏图较好）；对有向图而言，邻接表是以顶点  $i$  为弧尾的弧，逆邻接表是以顶点  $i$  为弧头的弧，邻接表可以求每个顶点的出度，而逆邻接表求每个顶点的入度。
  - ✓ 十字链表：有向图的一种链式存储结构（了解）
  - ✓ 邻接多重表：无向图的另一种链式存储结构（了解）
3. 图的基本操作
  - ✓ **深度优先搜索**：从一个顶点开始，访问该结点，依次从未被访问的邻接点出发，深度优先遍历图。借助于数据结构栈
  - ✓ **广度优先搜索**：从一个顶点  $v$  开始，访问该结点，依次访问  $v$  的各个未曾访问过的邻接点，然后分别从这些邻接点出发一次访问它们的邻接点，并是“先被访问的顶点的邻接点”先于“后被访问的顶点的邻接点”被访问。借助于数据结构队列。
4. 图的连通性问题
  - ✓ **最小生成树**：
    - 特点： $n$  个顶点， $n-1$  条边，边的权值和最小；
    - 注重构造过程：**Prim、Kruscal 算法**；
    - 应用：工程的造价最小。
5. 有向无环图及应用
  - ✓ 拓扑排序：
    - 概念：由某个集合上的一个**偏序**（集合中仅有部分成员之间可比较）得到该集合上的一个**全序**（集合中全体成员之间均可比较）的操作过程（人为的给成员之间加一些关系：谁比谁大，谁领先于谁）。
    - 排序的方法：
    - **应用**：判断图中有没有环。

- ✓ 关键路径
  - 概念：源点到汇点到最长路径。
  - 求关键路径：顶点表示事件，边表示活动。活动最早=最迟开始时间，P103~P108
  - 应用：估算整个工程的完工进度

## 第五部分、查找和排序（选择、综合）

### 1. 查找

- ✓ 静态查找表：
    - 顺序表的查找：顺序查找
    - 有序表的查找：折半查找
    - 索引顺序表的查找：分块查找
  - ✓ 动态查找表：
    - 二叉排序树（二叉查找树）：特点：（左子树的值都小于根结点的值，右子树的值都大于根结点的值，会构造二叉排序树）；中序遍历二叉排序树可得到一个关键字的有序序列。操作：插入（只改动指针的值）和删除（叶子结点、只有左子树或者只有右子树、左右子树都有）。计算 ASL（针对具体的二叉排序树）。和折半查找类似，与给定值比较的关键字个数不超过树的深度，但折半查找长度为  $n$  的表的判定树是唯一的，二含有  $n$  个结点的二叉排序树却不唯一。
  - ✓ 哈希查找
    - 定义：根据设定的哈希函数和处理冲突的方法，将一组关键字映像到一个有限的连续的地址集（区间）上，并以关键字在地址集中的“像”作为记录在表中的存储位置，这种表称为哈希表。
    - 构造哈希函数的方法：6 种（直接定址法；数字分析法；平方取中法；折叠法；除留余数法；随机数法）→好的哈希函数：值映射到表中，分布比较均匀。
    - 处理冲突的方法：4 种（开放定址法： $(H(key) + di) \bmod m$   $di$  线性探测，二次探测，伪随机探测。再哈希法。链地址法。建立一个公共溢出区）
    - 掌握程度：根据已知的哈希函数和处理冲突的方法，对具体值创建哈希表，也会利用哈希表进行查找。
- ### 2. 排序（内部排序）
- ✓ 概念
  - ✓ 排序算法的稳定性
  - ✓ 具体的排序算法：掌握每一类、每一种算法的排序特点；针对一个无序序列进行各种算法的实际操作
    - 插入类：直接插入排序；折半插入排序；希尔排序
    - 交换类：起泡排序；快速排序
    - 选择类：简单选择排序；堆排序（建初始堆；输出堆顶元素后调整堆）
    - 归并排序：2 路归并