



高等院校计算机教材系列

# 算法与数据结构 考研试题精析

第2版

陈守孔 胡潇琨 李玲 编著



机械工业出版社  
China Machine Press

本书第2版收集了自1992年以来国内60余所重点高校和科学院、所300多套硕士研究生入学“算法与数据结构”考试试卷的1600多道试题，并给出了参考答案和分析。

## 本书特点

- 本书包含的题量大、题型多，分为选择题、填空题、应用题、算法设计题。
- 主要选择名牌院校的试题，覆盖教材内容，同时偏重于重点章节。
- 指明试题出处，使读者如身临考试现场，以便有针对性地参考。
- 给出了参考答案，方便读者快捷而高效地学习。

## 本书的配套书：



《算法与数据结构（C语言版）》

书号：7-111-14620-4

定价：28.00元



《算法与数据结构实验与习题》

书号：7-111-14825-8

定价：19.00元

上架指导：计算机/算法与数据结构

投稿热线：(010) 88379604  
购书热线：(010) 68995259, 68995264  
读者信箱：hzsj@hzbook.com

华章网站 <http://www.hzbook.com>

网上购书：[www.china-pub.com](http://www.china-pub.com)



ISBN 978-7-111-15159-3



9 787111 151593

ISBN 978-7-111-15159-3  
定价：42.00元

高等院校计算机教材系列

# 算法与数据结构 考研试题精析

第2版

陈守孔 胡满琨 李玲 编著



机械工业出版社  
China Machine Press

本书收集了自1992年以来国内60余所重点高校和科学院、所300多套硕士研究生入学“算法与数据结构”考试试卷的1600多道试题，并给出了参考答案和分析。

本书可以作为高等院校计算机及相关专业学习数据结构课程的参考书，对于报考相关专业硕士研究生的考生也极具参考价值，同时也适用于讲授该课程的教师以及自修该课程的其他人员。

本书可以与《算法与数据结构（C语言版）》（ISBN 7-111-14620-4）教材配套使用。

**版权所有，侵权必究。**

**本书法律顾问 北京市展达律师事务所**

### 图书在版编目（CIP）数据

算法与数据结构考研试题精析 第2版/陈守孔，胡潇琨，李玲编著 —北京：机械工业出版社，2007.7

（高等院校计算机教材系列）

ISBN 978-7-111-15159-3

I. 算… II. ①陈 ②胡 ③李 III ①电子计算机-计算方法-研究生-入学考试-解题 ②数据结构-计算方法-研究生-入学考试-解题 IV TP311.12-44

中国版本图书馆CIP数据核字（2007）第069385号

机械工业出版社（北京市西城区白万庄大街22号 邮政编码 100037）

责任编辑：李东震

北京牛山世兴印刷厂印刷 新华书店北京发行所发行

2007年7月第2版第1次印刷

184mm×260mm·27.75印张

定价：42.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换  
本社购书热线：（010）68326294

## 第2版前言

自《算法与数据结构考研试题精析》第1版出版以来，得到了读者的好评。为了反映近几年考研试题的变化，更好地为读者服务，编者对本书进行了全面修订。

首先是对试题进行了增删，删除了400多道试题，替换了200多道试题，按编号计算再版试题共1659题，其中算法设计327题。不再设立“类似本题的叙述”这部分内容，每题都是单独编号。对参考答案进行了审核，尽量做到答案准确、简练。

在准备再版的资料时，编者再次印证了“数据结构的考研试题重复量很大”的结论。在收集的近几年的试题中，绝大部分试题都可在本书第1版中找到，尤其是基础知识题，有些试题的叙述完全一样。由此可以看出，弄懂了本书的试题，无疑将对考研有很大帮助。

通过整理这几年的试题，编者还发现，几乎所有院校都突出了对基础知识的考查。本书中各章的第一到第四部分就属于基础知识。过去，个别院校的试题过于侧重算法设计，而没有基本概念和基本知识。现在大多数院校的硕士研究生考试将专业课整合为一张试卷（占150分）（少数院校仍单独考核数据结构），其中包括两门课程或三门课程的内容。数据结构所占的分数少则50分，多则90分，并且基础知识占多数，一道算法设计题占20分以上的现象已很少见。所以，编者希望读者，特别是考研的学生，应该加强基础知识的学习。

为了节省篇幅，避免在每道试题解答中重复定义所用数据结构，在本书附录中将给出所用的数据结构，试题解答中将直接使用。

编者欣喜获悉，许多教师将本书作为教学参考书和考试的题库，考研学生通过学习本书大大提高了考研的成绩，我们期望本书在教学中发挥更大作用。

尽管我们作了很大努力，但由于水平有限，书中难免会有缺点错误，恳请读者批评指正。

编者

2007年3月

## 第1版前言

“算法与数据结构”课程是高等学校计算机科学与技术专业的一门重要的综合专业基础课，近年来也成为非计算机专业的必修课或选修课。在以往的硕士研究生入学考试中，该课程是计算机类专业的必考科目，也是相关专业的考试科目。

编者多年来在大学讲授“算法与数据结构”课程。在教学中感到，学生理解课程的概念和书本知识并不困难，一旦涉及解决具体问题，特别是编制算法，往往无从着手。为了加强学生对本课程基本概念和基础知识的理解，特别是加强对编写算法的训练，我们编写了本书。

本书从编排上分三部分。第一部分简要复习数据结构各章的重点，第二部分是编者收集的自1992年以来国内68所重点高校和科学院、所300多套硕士研究生入学“算法与数据结构”考试试卷的1800多道试题，第三部分给出了参考答案和分析。

本书的各章名称与《算法与数据结构（C语言版）》教材相同。每章分选择题、判断题、填空题、应用题和算法设计题五部分。前四类题属于基础知识。选择题多是单选，也有少数是多选，编者直接给出参考答案；判断题是判断对错，除给出参考答案外，还对个别题给予了解释；填空题有概念填空、计算填空，值得注意的是有些院校的算法（程序）填空，即填上几个关键语句，使之成为完整算法（程序），这类题要求较高；应用题有的回答基本概念和基础知识，较多的是手工模拟算法，这部分占的比例较大；算法设计是本书重点，占的篇幅最大，除比较简单的题外，多数题都按题目分析、算法设计、算法讨论三部分展开。算法设计中除题目要求必须用PASCAL语言描述的外，一律用类C语言描述。算法描述中涉及的类型定义和数据结构基本取自本书的配套教材《算法与数据结构（C语言版）》，为节省篇幅，本书不再重新定义而直接使用。

试题的选取原则是：覆盖教材各章节，兼顾重点章节；主要选名牌院校的考题；同类型试题解答一个，列出类似试题，多数未作解答。列出类似题的目的之一，是引起学生对该类题的重视，考题重复率高也从侧面说明了该类题的重要性。由于本书收集的是全国各高校和科学院、所的试题，加之各校教材不同，所以在题目的叙述上有很大差异。甚至所用名词、概念也不相同。语言描述上有PASCAL语言、类C语言、框图和伪码等，叙述及算法描述中的大小写不是很统一。我们尽量尊重原题，为保持本书风格大体一致，对部分术语进行了统一。另外，在每道题后都注明了题目出处，例如【清华大学 1997 三（10分）】的含义是本题选自清华大学1997年硕士研究生数据结构试题第三题，试题分数是10分，有的还指出大题中的小题。对于类似题，个别的也作了简单解答。

试题也按教材分11章列出。但试题内容具体分到哪章，其划分并不唯一。例如，线性表的问题，可以放在第2章，也可能因其用顺序存储结构实现使用了数组而放到第5章，也可能因排序而放到第10章，甚至因用顺序查找而放到第9章。本书各章相互独立，在使用本书时，可以顺序学习，也可以根据需要直接选择某章。

为了增大本书的信息量，在保持算法易读性的前提下，尽量使用多语句行，尽量减少图（使用表格代替图形）。

本书是很多人的劳动结晶。计算机学院的学生购买了试题，并进行了文字输入。宁方美、田相庆、庞圣波、王景波等同学对输入的试题进行了校对。范策、孟佳娜、卢云宏等老师对算法提出了一些建议，编者对所有帮助编写本书的同志表示衷心的感谢。在成书过程中，还得到了机械工业出版社的支持和帮助，在此表示衷心的感谢。

胡潇琨老师编写了本书的第10章，并做了试题归类等工作。李玲老师编写了本书的第1章，绘制了大量图表，并核查了部分算法。本书中除第1章、第10章外的其余内容均由陈守孔老师编写。

我们尽全力保证本书的质量，但由于水平有限，加之时间紧张，书中肯定会有缺点和错误，特别是算法的编写很难保证是优化的。编者诚恳地期望读者给予批评指正。

编者

2004年4月于烟台大学

# 目 录

## 第一部分 复习纲要

第1章 概论 .....	2
第2章 线性表 .....	2
第3章 栈和队列 .....	3
第4章 串 .....	4
第5章 数组和广义表 .....	4
第6章 树 .....	4
第7章 图 .....	5
第8章 动态存储管理 .....	6
第9章 集合 .....	6
第10章 排序 .....	7
第11章 文件 .....	8

## 第二部分 试题部分

第1章 概论 .....	10
第2章 线性表 .....	18
第3章 栈和队列 .....	40
第4章 串 .....	54
第5章 数组和广义表 .....	62
第6章 树 .....	81
第7章 图 .....	114

第8章 动态存储管理 .....	139
第9章 集合 .....	141
第10章 排序 .....	163
第11章 文件 .....	189

## 第三部分 参考答案

第1章 概论 .....	194
第2章 线性表 .....	199
第3章 栈和队列 .....	231
第4章 串 .....	252
第5章 数组和广义表 .....	263
第6章 树 .....	285
第7章 图 .....	332
第8章 动态存储管理 .....	364
第9章 集合 .....	367
第10章 排序 .....	399
第11章 文件 .....	427
附录 本书中使用的数据结构的类型 定义 .....	431
参考文献 .....	437



## 第一部分

---

# 复习纲要

## 第1章 概 论

### 一、内容提要

1. 数据结构研究的内容。
2. 基本概念：数据、数据元素、数据对象、数据结构、数据类型、抽象数据类型、多型数据类型。
3. 算法的定义及五个特征。
4. 算法描述的方法：类C语言。
5. 算法设计要求。
6. 算法分析、时间复杂度和空间复杂度。

### 二、知识点

1. 数据的三个层次：数据、数据元素、数据项。
2. 数据结构的“三要素”：逻辑结构、物理（存储）结构及在这种逻辑结构上所定义的操作（运算）。
3. 抽象数据类型的定义、表示和实现方法。
4. 类C语言书写规范。
5. 时间复杂度，最佳（最坏、平均）时间复杂度。常用计算语句频度来估算算法的时间复杂度，算法时间复杂度量级（常量、线性、平方等）。有时要求分析语句执行的次数，有时要求给出语句执行次数的数量级。
6. 空间复杂度，算法原地工作指算法所需辅助空间是常量。

## 第2章 线 性 表

### 一、内容提要

1. 线性表是元素间约束力最强的一类数据结构，非空线性表第一个元素无前驱只有后继，最后一个元素无后继只有前驱，其余每个元素均有唯一前驱和唯一后继。
2. 线性表的逻辑结构定义，对线性表的操作。
3. 线性表的存储结构：顺序存储结构和链式存储结构。
4. 线性表的操作在两种存储结构中的实现。
5. 一元多项式的线性表表示方法，高次（稀疏）多项式的抽象数据类型定义、表示和加法的实现。

### 二、知识点

1. 线性表的逻辑结构，指线性表的数据元素间存在着线性关系。在顺序存储结构中，元素存储的先后位置反映出这种逻辑关系，而在链式存储结构中，是靠指针来反映这种逻辑关系的。
2. 顺序存储结构用向量（一维数组）表示，给定下标，可以存取相应元素，属于随机存取存储结构。

3. 尽管“只要知道某结点的指针就可以存取该元素”，但因链表的存取都需要从头指针开始，顺链而行，故链表不属于随机存取结构。

4. 链表是本章学习的重点和难点。要理解头指针、头结点、首元结点和元素结点的差别。头结点是在插入、删除等操作时，为了算法的统一而设立的（若无头结点，则在第一元素前插入元素或删除第一元素时，链表的头指针总在变化）。掌握通过画出结点图来进行链表的生成、插入、删除、遍历等操作的方法。对链表（不包括循环链表）的任何操作，均要从头结点开始，头结点的指针具有标记作用，故头指针往往被称为链表的名字，如链表 $la$ ，既指出链表的名字是 $la$ ，同时也指出链表头结点的指针是 $la$ 。

5. 链表操作中应注意不要使链意外“断开”。因此，若在某元素前插入一个元素或删除某元素，必须知道该元素的前驱结点的指针。

6. 从时间和空间复杂度的角度综合比较线性表在顺序和链式两种存储结构下的特点。

7. 静态链表是又一重点和难点，应和链表进行对比理解。例如，链表 $la$ 在有头结点的情况下，第一元素可表示为 $la \rightarrow next$ ，而在静态链表 $sa$ 中，静态链表也常用下标0作“头结点”，其第一元素结点是 $sa[0].next$ ；相对 $p=p \rightarrow next$ ，有 $i=sa[i].next$ 来找到第 $i$ 个元素的后继；对链表用 $p==null$ 判断是否到尾，而静态链表用 $i==-1$ 来判断链表是否结束。树的双亲表示法、图的邻接表表示法、表插入排序、链式基数排序、地址排序等都是静态链表的应用。

## 第3章 栈和队列

### 一、内容提要

1. 从数据结构角度讲，栈和队列属于线性结构，其操作是线性表操作的子集，是操作受限的线性表。但从数据类型的角度看，它们是和线性表大不相同的重要抽象数据类型。

2. 栈的定义及操作。栈是只准在一端进行插入和删除操作的线性表，该端称为栈的顶端。

3. 栈的顺序和链式存储结构，及在这两种结构下实现栈的操作。

4. 栈的应用：表达式求值，过程调用，递归过程及消除递归。

5. 队列的定义及操作，队列的删除在一端（头），而插入则在另一端（尾）。因此在两种存储结构中，一般都需要队头和队尾两个指针。

6. 链队列空的条件是首尾指针相等，而循环队列满的条件的判定，则有牺牲一个单元和设标记两种方法。

### 二、知识点

1. 栈和队列操作在两种存储结构下的实现，注意因栈在一端操作，通常链栈不设头结点。

2. 中缀表达式转成前缀、后缀表达式，并对两种表达式求值。

3. 用递归解决的问题：问题的定义是递归的，数据结构是递归的，以及问题的解法是递归的，掌握典型问题的算法。将递归算法转为非递归算法，特别是尾递归的消除。

4. 对仅剩一个元素的链队列删除元素时的处理（令队尾指针指向队头）。特别是仅设尾指针的循环链队列的各种操作的实现。

5. 循环队列中队列空用队头指针等于队尾指针来判断，队列满则可用牺牲一个单元（教材中所示）及设标记方法。这里特别注意入队、出队和求元素个数等操作中的取模运算。

6. 在后续章节中多处有栈和队列的应用，如串中心对称的判定，二叉树遍历的递归和非递

归算法，图的深度优先遍历等都用到栈，而树的层次遍历、图的广度优先遍历等则用到队列。

## 第4章 串

### 一、内容提要

1. 串是数据元素为字符的线性表，串的定义及操作。
2. 串的基本操作，用串的基本操作来编写算法求串的其他操作。
3. 串的存储结构，因串是数据元素为字符的线性表，所以存在“结点大小”的问题。静态和动态（块链结构、堆结构）存储结构的优缺点。
4. 朴素模式匹配算法及改进（KMP）算法。

### 二、知识点

1. 本章重点是用串的基本操作编写串的其他操作（如 $index$ 、 $replace$ 等）和串的模式匹配。
2. 了解KMP算法的推导过程，熟练掌握手工描述求匹配串的 $next$ 和 $nextval$ 函数值。
3. 尽管朴素的模式匹配的时间复杂度是 $O(m*n)$ ，KMP算法的时间复杂度是 $O(m+n)$ 。但在一般情况下，前者实际执行时间近似 $O(m+n)$ ，因此至今仍被采用。KMP算法仅在主串与模式串存在许多“部分匹配”时才显得比前者快得多，其主要优点是主串不回溯。
4. 串操作在各种存储结构下的实现。

## 第5章 数组和广义表

### 一、内容提要

1. 数组的逻辑结构定义及存储。
2. 稀疏矩阵（含特殊矩阵）的存储及运算。
3. 广义表的定义以及存储。
4. 广义表运算的递归算法。

### 二、知识点

1. 数组（主要是二维）在以行序为主和列序为主的存储中的地址计算方法。
2. 特殊矩阵在压缩存储时的下标变换公式。
3. 稀疏矩阵的三元组表存储结构及矩阵转置的算法。
4. 稀疏矩阵的十字链表存储方法及十字链表生成算法。
5. 广义表的HEAD和TAIL运算。
6. 给定广义表画出其存储结构。
7. 通过广义表的递归模型，掌握如何编写递归算法。

## 第6章 树

### 一、内容提要

1. 树是复杂的非线性数据结构，树、二叉树的递归定义、基本概念、术语。

2. 二叉树的性质、存储结构。
3. 二叉树的遍历算法（递归、非递归）。
4. 线索二叉树。
5. 树的存储结构，树、森林的遍历及和二叉树的相互转换。
6. 二叉树的应用：表达式求值、判定问题及哈夫曼树和哈夫曼编码。

## 二、知识点

1. 二叉树性质及证明方法，并把这种方法推广到K叉树。
2. 二叉树遍历的递归算法，教材中介绍了三种（前序、中序、后序）方法，另三种也应会用（如按降序输出二叉排序树中结点的值）。前序和中序的非递归遍历（后序非递归遍历的特殊性）。遍历是基础，由此导出许多实用的算法，如求二叉树的高度和各结点的层次数以及度为0、1、2的结点数，二叉树的相似、全等、复制等。
3. 由二叉树的遍历的前序和中序序列或后序和中序序列可以唯一确定一棵二叉树，要会手工模拟及编写算法。由前序和后序序列不能唯一确定一棵二叉树。
4. 二叉树线索化的实质是建立结点在相应序列中与其前驱和后继之间的直接联系。要注意两个定语：何序（前序、中序和后序）下进行何种线索化（全线索化、前驱线索化和后继线索化），并会求某结点在某种线索化下的前驱和后继。
5. 完全二叉树的高度及其双亲与子女的编号关系，二叉树顺序存储结构和二叉链表存储结构的相互转换的算法。
6. 树的双亲表示法和孩子兄弟表示法间的相互转换算法。
7. 树、森林和二叉树间的相互转换（三步曲：“连线”、“切线”和“旋转”）。
8. 哈夫曼树的定义、构造及求哈夫曼编码。一般说，哈夫曼树的形态不是唯一的。

## 第7章 图

### 一、内容提要

1. 图的定义，概念、术语及基本操作。
2. 图的存储结构，特别是邻接矩阵和邻接表。
3. 图的深度优先遍历和广度优先遍历。
4. 图的应用（连通分量，最小生成树，拓扑排序，关键路径，最短路径）。

### 二、知识点

1. 基本概念中，完全图、连通分量、生成树和邻接点是重点。
2. 图是复杂的数据结构，也有顺序和链式两种存储结构：数组表示法（重点是邻接矩阵）和邻接表。这两种存储结构对有向图和无向图均适用。十字链表是有向图的另一种表示方法，将有向图的邻接表和逆邻接表合一。邻接多重表是无向图邻接表的改进，将边结点的数量减少一半（边结点数等于边数）。
3. 图的遍历是图的各种运算的基础，应熟练掌握图的深度、广度优先遍历，手工模拟图的遍历中栈和队列指针状态的变化。
4. 在（强）连通图中，主过程一次调用深（宽）度优先遍历过程（dfs/bfs），即可遍历完

全部顶点，故可以用此方法求出连通分量的个数，要会画出遍历中形成的深（宽）度优先生成树和生成森林。

5. 连通图的最小生成树通常不是唯一的，但最小生成树边上的权值之和是唯一的。应熟练掌握Prim和Kruskal算法，特别是手工分步模拟生成树的生成过程。

6. 拓扑排序是在有向图上对入度（先、后）为零的顶点的一种排序，通常结果不唯一。关键路径是在拓扑有序的前提下求出来的从源点到汇点的最长路径。应能掌握这两种算法，并熟练手工模拟。理解“减少关键活动时间能缩短工期”，是指该活动为所有关键路径所共有，且减少到尚未改变关键路径的前提下才有效。

7. 用拓扑排序和深度优先遍历都可判断图是否存在环路，对有向图环路的判断更复杂。

8. 从单源点到其他顶点以及各个顶点间的最短路径问题，掌握Dijkstra和Floyd算法，并能手工熟练模拟。掌握用求最短路径来解决的应用问题（如医院设在哪个村庄等）。

## 第8章 动态存储管理

### 一、内容提要

1. 动态存储管理指的是在用户需要时给分配内存，而在用户结束使用时，系统要收回用户所占空间。

2. 可利用空间表的三种结构形式：结点固定大小；结点大小分几种规格；结点任意大小。

3. 可利用空间表的两种组织形式：目录表，链表。

4. 可利用空间表的分配方式：首次拟合法，最佳拟合法，最差拟合法。要了解每种分配方式的使用范围。

5. 可利用空间表的分配和回收的两种基本实现方法：边界标识法，伙伴系统。特别要掌握伙伴系统中伙伴地址的计算方法。

6. 无用单元回收和紧缩存储的概念。

### 二、知识点

1. 概念：可利用空间表及分配方式，紧缩存储，伙伴系统等。

2. 边界标识法的分配及回收算法。

3. 伙伴系统的分配及回收算法，伙伴地址的求法。

## 第9章 集 合

### 一、内容提要

1. 本章介绍的集合数据结构，是元素间约束力最差的数据结构：元素间的唯一关系是元素共在同一个集合中，元素间不存在任何逻辑关系。

2. 集合的操作：查找，检索，插入，删除，成员关系判断。

3. 静态查找表：顺序表，有序表，静态树表，索引顺序表。

4. 动态查找表：二叉排序树，平衡二叉树，B-树，B+树，键树。

5. 哈希表。

## 二、知识点

1. 集合这种数据结构，因元素间关系非常松散，其操作需借助其他数据结构来实现。本章列举了三种方法（静态查找表，动态查找表，哈希表）实现集合的运算。

2. 顺序表因设置了监视哨使查找效率大大提高。有序表的平均查找长度不超过树的深度，其判定树是唯一的。索引顺序查找综合了上述二者的优点，既能较快速地查找，又能适应动态变化的要求。顺序查找、有序表查找、索引顺序查找的时间复杂度分别为 $O(n)$ 、 $O(\log n)$ 、 $O(n^{1/2}+1)$ （在 $s=n^{1/2}$ 的情况下）。

3. 二叉排序树的形态取决于元素的输入顺序，二叉排序树在最差情况下形成单支树，平均查找长度是 $O(n)$ ，最佳情况下是 $O(\log n)$ ，平均情况不大于 $1+4 \log n$ 。按中序遍历可得到结点的有序序列，应熟练掌握其建立、查找、插入和删除算法。

4. 最佳二叉排序树是平均查找长度最短的二叉排序树，平衡二叉树是介于最佳二叉排序树和一般二叉排序树间的折中，应熟练掌握手工绘制平衡二叉树。

5. B-树是多路平衡外查找树，用于文件系统。要能手工模拟B-树插入和删除关键字使B-树增高和降低。会推导B-树的平均查找长度。

6. 键树中每个结点是关键字的一个字符，该树是有序树（同层兄弟间从左到右递增，最小是结束符号\$）。

7. 哈希表是集合的又一表示方法，根据选定的哈希函数和处理冲突的方法，将关键字映像到哈希表中。一般来说冲突是不可避免的。

8. 哈希表中关键字的查找只能用哈希函数来计算，不能顺序查找，也不能折半查找。在闭散列法解决冲突的情况下，元素删除也只能作标记，不能物理地删除。理想情况下，哈希表的平均查找长度是 $O(1)$ ，优于其他查找方法。

## 第10章 排 序

### 一、内容提要

1. 排序的定义，排序可以看作是线性表的一种操作。
2. 排序的分类，稳定排序与不稳定排序的定义。
3. 插入排序（折半插入、二路插入、表插入、希尔插入）。
4. 交换排序（起泡排序、快速排序）。
5. 选择排序（直接选择排序、树形选择排序、堆排序）。
6. 归并排序和基数排序。
7. 外部排序指待排序文件较大，内存一次存放不下，需存放在外部介质的文件的排序。
8. 为减少平衡归并中外存读写次数所采取的方法：增大归并路数和减少归并段个数。
9. 利用败者树增大归并路数。
10. 利用置换选择排序增大归并段长度来减少归并段个数。
11. 由长度不等的归并段进行多路平衡归并，需要构造最佳归并树。
12. 磁带的多步归并排序。

### 二、学习要点

1. 各种排序所基于的基本思想。

2. 在“最好”和“最差”情况下排序性能的分析，是否是稳定排序的结论。
3. 插入排序的基本思想是基于假定待排序文件第一个记录有序，然后从第二个记录起，依次插入到已排好序的有序子文件中，直到整个文件有序。从减少比较次数和移动次数进行了各种改进，产生了折半、二路、表插入、希尔等一系列插入排序。
4. 交换排序基于相邻记录比较，若逆序则进行交换。起泡排序和快速排序是交换排序的例子，快速排序是目前最快的内部排序法。应采用“三者取中”法防止其性能退化。
5. 直接选择排序、树形选择排序、堆排序是选择排序的例子。堆排序较为重要，其最差性能比快速排序的最差性能好 ( $O(\log n)$ )。
6. 归并排序、基数排序及时间复杂度  $O(n^2)$  的排序为稳定排序，而希尔排序、快速排序、堆排序等时间性能好的排序方法是不稳定排序（但特别注意，直接选择排序虽然时间复杂度是  $O(n^2)$ ，但却是不稳定排序）。
7. 熟悉外部排序的两个阶段，归并过程。
8. 掌握外部排序过程中进行外存读/写次数的计算方法。
9. “胜者树”增大归并路数不能减少外存读写次数，“败者树”则可以。掌握败者树的建立及归并算法。
10. 熟悉置换选择排序的过程，理解它能得到平均长度为工作区两倍的初始归并段。
11. 熟练掌握最佳归并树的构造方法及该过程中对外存读/写次数的计算方法。
12. 了解磁带多步归并的特点，熟悉归并过程及设置虚段的方法及归并过程所需外存读/写次数的计算方法。

## 第11章 文 件

### 一、内容提要

1. 文件基本概念，文件分类，记录的逻辑和物理结构。
2. 顺序文件，索引文件（ISAM和VSAM），直接存取文件（散列文件）。
3. 多关键字文件：多重表文件和倒排文件。

### 二、知识点

各类文件的特点，构造方法及如何实现检索、插入和删除等操作。



## 第二部分

---

# 试 题 部 分