



学府考研
中国优秀高端教育品牌


理工社®

2016 计算机考研 综合考点精讲 与复习指导全书

◎主编 杨航空

◎策划 学府考研计算机命题研究组

- 复习要点 建构宏观知识网络
- 考点精讲 全面突破考纲难点
- 典型例题 独特剖析解题技巧
- 真题演练 明确把握考题规律

赠

本书免费同步讲解视频，
扫描书内二维码或登录：

<http://www.kaoshidian.com>

 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS

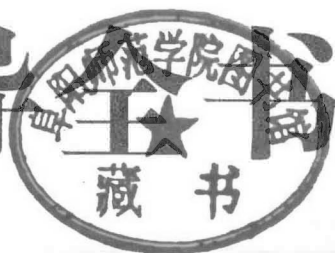


学府
中国 优秀 高端



2016 计算机考研

综合考点精讲 与复习指导



◎主编 杨明坚

◎策划 学府考研计算机命题研究组

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

图书在版编目 (CIP) 数据

计算机考研综合考点精讲与复习指导全书/杨航空主编. —北京: 北京理工大学出版社, 2015. 7
ISBN 978 - 7 - 5682 - 0600 - 6

I. ①计… II. ①杨… III. ①电子计算机 - 研究生 - 入学考试 - 自学参考资料 IV. ①TP3

中国版本图书馆 CIP 数据核字 (2015) 第 099222 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)
(010) 82562903 (教材售后服务热线)
(010) 68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 三河市刚利印务有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 38.5

字 数 / 1045 千字

版 次 / 2015 年 7 月第 1 版 2015 年 7 月第 1 次印刷

定 价 / 69.50 元

责任编辑 / 陈莉华

文案编辑 / 陈莉华

责任校对 / 周瑞红

责任印制 / 边心超

前言

Preface

《计算机考研综合考点精讲与复习指导全书》以教育部考试中心发布的《计算机专业基础综合考试大纲》为蓝本,严格按照全国计算机学科专业考研大纲的内容,对大纲所涉及的知识点进行集中梳理,力求内容精练、重点突出、深入浅出。对大纲中的重点、难点内容进行讨论,精选了名校历年考研真题,并给出详细解题思路和算法,力求达到讲练结合、灵活掌握、举一反三的功效。通过《计算机考研综合考点精讲与复习指导全书》可大大提高考生的复习效果,达到事半功倍的复习效率。《计算机考研综合考点精讲与复习指导全书》可作为考生参加计算机专业研究生入学考试的备考复习用书,也可作为计算机专业的学生学习计算机相关课程的辅导用书。

全书在章节安排上与考研大纲保持一致,分为数据结构、计算机组成原理、计算机操作系统、计算机网络四大部分。每部分中的章节由复习要点、考点精讲、典型例题和真题演练组成。

(1) **复习要点**——使考生明确各章的重点、难点及常考点,弄清各知识点之间的相互联系,以及多年考试中本章节的出题情况,以便对本章内容有一个全局性的认识和把握。

(2) **考点精讲**——本部分参考当前国内最权威的大学教材,对大纲所要求的知识点进行了全面、准确地阐述,以加深考生对基本概念和原理等重点内容的理解和正确应用。本部分讲解考点明确、重点突出、层次清晰、简明实用。

(3) **典型例题**——通过对经典例题的分析教会考生分析问题、解决问题的方法和技巧。通过大量练习题,使考生做到学练结合,更好地巩固所学知识,提高实战能力。

(4) **真题演练**——真题演练中给出了2009—2015年的部分计算机专业基础综合真题,考生可以通过对真题的练习,掌握考试的出题形式及答题技巧。

本书具有以下特点:

(1) 依据最新考试大纲编写,对考纲要求的知识点进行全面归纳,并对重点和难点做了标注。

(2) 使用层次树状图对知识点进行归纳,方便考生记忆重点考点,梳理知识脉络,形成对计算机基础综合科目宏观上的把握。

(3) 以插图的形式讲解知识点,方便考生形象地理解计算机基础综合考试中涉及的原理。

(4) 选取了计算机专业基础综合考试中的部分真题,通过真题与知识点的紧密结合,考生能够更明确地了解考点,把握考试规律。

本书适合基础复习阶段和强化复习阶段使用,在基础复习结束之后,建议考生认真研究书中的真题,将真题弄通,能够举一反三,并结合重要考点多加练习。

在复习初期,我们建议读者最好对照本书的配套练习《计算机考研通关必做1200题》一书进行复习,学习完一章的知识即做该章相应的练习,以检测复习效果,帮助理解和掌握考点。《计算机考研综合考点精讲与复习指导全书》可贯穿复习始终,前期可以作为同步训练,后期用于强化训练。

通过有的放矢地辅导和练习,加上考生的刻苦努力,我们相信,梦想定会变成现实。

预祝各位考生在未来的研究生考试中笑傲考场,书写自己的传奇!

目录

Contents

第一部分 数据结构

第一章 线性表	3
第一节 线性表的逻辑结构	3
第二节 线性表的顺序存储结构	4
第三节 线性表的链式存储结构	7
本章经典真题及解析	19
第二章 栈、队列和数组	27
第一节 栈与队列	27
第二节 数组	38
本章经典真题及解析	43
第三章 树与二叉树	46
第一节 树的概念	46
第二节 二叉树	47
第三节 树和森林	60
第四节 树的应用	64
本章经典真题及解析	76
第四章 图	81
第一节 图的基本概念	81
第二节 图的存储及基本操作	84
第三节 图的遍历	89
第四节 图的基本应用	92
本章经典真题及解析	103
第五章 查找	108
第一节 查找的基本概念	108
第二节 顺序查找	109
第三节 折半查找	112
第四节 分块查找	115
第五节 B ⁻ 树和 B ⁺ 树	116

第六节 散列表查找	123
第七节 字符串匹配	130
本章经典真题及解析	134
第六章 排序	137
第一节 排序的基本概念	137
第二节 插入排序	138
第三节 冒泡排序	140
第四节 简单选择排序	142
第五节 希尔排序	143
第六节 快速排序	145
第七节 堆排序	146
第八节 二路归并排序	149
第九节 基数排序	150
第十节 外部排序	153
第十一节 各种内部排序算法的比较	155
本章经典真题及解析	158

第二部分 计算机组成原理

第一章 计算机系统概述	163
第一节 计算机的发展历程	163
第二节 计算机系统的层次结构	166
第三节 计算机的性能指标	172
本章经典真题及解析	174
第二章 数据的表示和运算	176
第一节 数制与编码	176
第二节 定点数的表示和运算	184
第三节 浮点数的表示和运算	189
第四节 算术逻辑单元 ALU	192
本章经典真题及解析	193
第三章 存储系统	198
第一节 存储器的分类	198
第二节 存储器的层次化结构	200
第三节 半导体随机存取存储器	204
第四节 只读存储器	208
第五节 主存储器与 CPU 的连接	209
第六节 双口 RAM 和多模块存储器	211

第七节 高速缓冲存储器 (Cache)	215
第八节 虚拟存储器	220
本章经典真题及解析	224
第四章 指令系统	236
第一节 指令格式	236
第二节 指令的寻址方式	241
第三节 CISC 和 RISC 的基本概念	245
本章经典真题及解析	247
第五章 中央处理器	252
第一节 CPU 的功能和基本结构	253
第二节 指令执行过程	256
第三节 数据通路的功能和基本结构	262
第四节 控制器的功能和工作原理	264
第五节 指令流水线	273
第六节 多核处理器的基本概念	282
本章经典真题及解析	282
第六章 总线	286
第一节 总线概述	286
第二节 总线仲裁	290
第三节 总线操作和定时	292
第四节 总线标准	294
本章经典真题及解析	296
第七章 输入/输出系统	299
第一节 I/O 系统基本概念	299
第二节 外部设备	301
第三节 I/O 接口 (I/O 控制器)	305
第四节 I/O 方式	307
本章经典真题及解析	317

第三部分 计算机操作系统

第一章 操作系统概述	323
第一节 操作系统的概念、特征、功能和提供的服务	323
第二节 操作系统的发展与分类	326
第三节 操作系统的运行环境	329
第四节 操作系统的体系结构	333
本章经典真题及解析	333

第二章 进程管理	335
第一节 进程与线程	335
第二节 进程同步	349
第三节 处理机调度	359
第四节 死锁	364
本章经典真题及解析	367
第三章 存储管理	377
第一节 内存管理基础	378
第二节 虚拟内存管理	391
本章经典真题及解析	402
第四章 文件管理	408
第一节 文件系统基础	408
第二节 文件系统实现	421
第三节 磁盘组织与管理	426
本章经典真题及解析	432
第五章 输入/输出管理	436
第一节 I/O 管理概述	436
第二节 I/O 核心子系统	445
本章经典真题及解析	455

第四部分 计算机网络

第一章 计算机网络概述	461
第一节 计算机网络概述	461
第二节 计算机网络体系结构与参考模型	464
本章经典真题及解析	471
第二章 物理层	473
第一节 通信基础	473
第二节 传输介质	482
第三节 物理层设备	488
本章经典真题及解析	490
第三章 数据链路层	492
第一节 数据链路层的功能	492
第二节 组帧	493
第三节 差错控制	495
第四节 流量控制与可靠传输机制	500
第五节 介质访问控制	505

第六节 局域网	513
第七节 广域网	521
第八节 数据链路层设备	525
本章经典真题及解析	529
第四章 网络层	533
第一节 网络层的功能	533
第二节 路由算法	536
第三节 IPv4	539
第四节 IPv6	552
第五节 路由协议	554
第六节 IP 组播	559
第七节 移动 IP	562
第八节 网络层设备	564
本章经典真题及解析	566
第五章 传输层	574
第一节 传输层提供的服务	574
第二节 UDP 协议	577
第三节 TCP 协议	578
本章经典真题及解析	587
第六章 应用层	590
第一节 网络应用模型	590
第二节 DNS 系统	592
第三节 FTP	595
第四节 电子邮件	596
第五节 WWW	599
本章经典真题及解析	602

第一部分 数据结构

2015 年的统考大纲对数据结构的考查目标定位为掌握数据结构的基本概念、基本原理和基本方法,掌握数据的逻辑结构、存储结构以及基本操作的实现;能够对算法进行基本的时间复杂度和空间复杂度的分析;能够运用数据结构的基本原理和方法进行问题的分析求解,具备采用 C、C++ 语言设计程序与实现算法的能力。

当然,考生也不必因此而专门复习一遍 C 或 C++ 程序设计,毕竟复习时间有限,而且数据结构要求的重点在于算法设计的能力,而不是编写代码的能力,因此,只要能用类似伪代码的形式把思路表达清楚就行,不用强求写出一个没有任何语法错误的程序。

数据(data)是信息的载体,是描述客观事物属性的数、字符以及所有能输入到计算机中并被计算机程序识别和处理的符号的集合。数据结构是指数据对象及其相互关系和构造方法。

数据结构中结点与结点之间的逻辑关系称为数据的逻辑结构,它是面向使用的。按数据的逻辑关系,数据结构可分为线性结构和非线性结构,其中非线性结构又可分为树形结构和图结构。一种特殊的数据结构称为集合。集合结构从逻辑上看,元素之间没有关系,但从实现上来看,它可以用线性结构或非线性结构来表示。

数据结构在计算机中的存储映像称为数据的存储结构。典型的存储方式有 4 种,包括顺序存储结构、链式存储结构、索引存储结构、散列存储结构。

算法与数据结构之间有密切的关系。算法建立在数据结构的基础上,选择合理的数据结构可有效地改进算法的效率。算法是过程性的,按输入—计算—输出的模式解决问题。对算法的分析只要着眼于它的时间代价和空间代价,需要熟练掌握的是大 O 表示法。



考查目标

- (1) 理解数据结构的基本概念;掌握数据的逻辑结构、存储结构及其差异以及各种基本操作的实现。
- (2) 在掌握基本的数据处理原理和方法的基础上,能够对算法进行设计与分析。
- (3) 能够选择合适的数据结构和方法进行问题求解。



考点变化

(1) 考查目标新增“具备采用 C、C++ 语言设计与实现算法的能力”。这一点的增加相信与 2009 年考试中出现了一道考查程序设计的题目有关系。

(2) 树的应用中去掉了“等价类问题”。“等价类问题”难度较高,在很多学校本科计算机数据结构的课程中都不对这一知识点有要求,因此,这一点的变化也体现了命题人对考生知识结构的考虑。

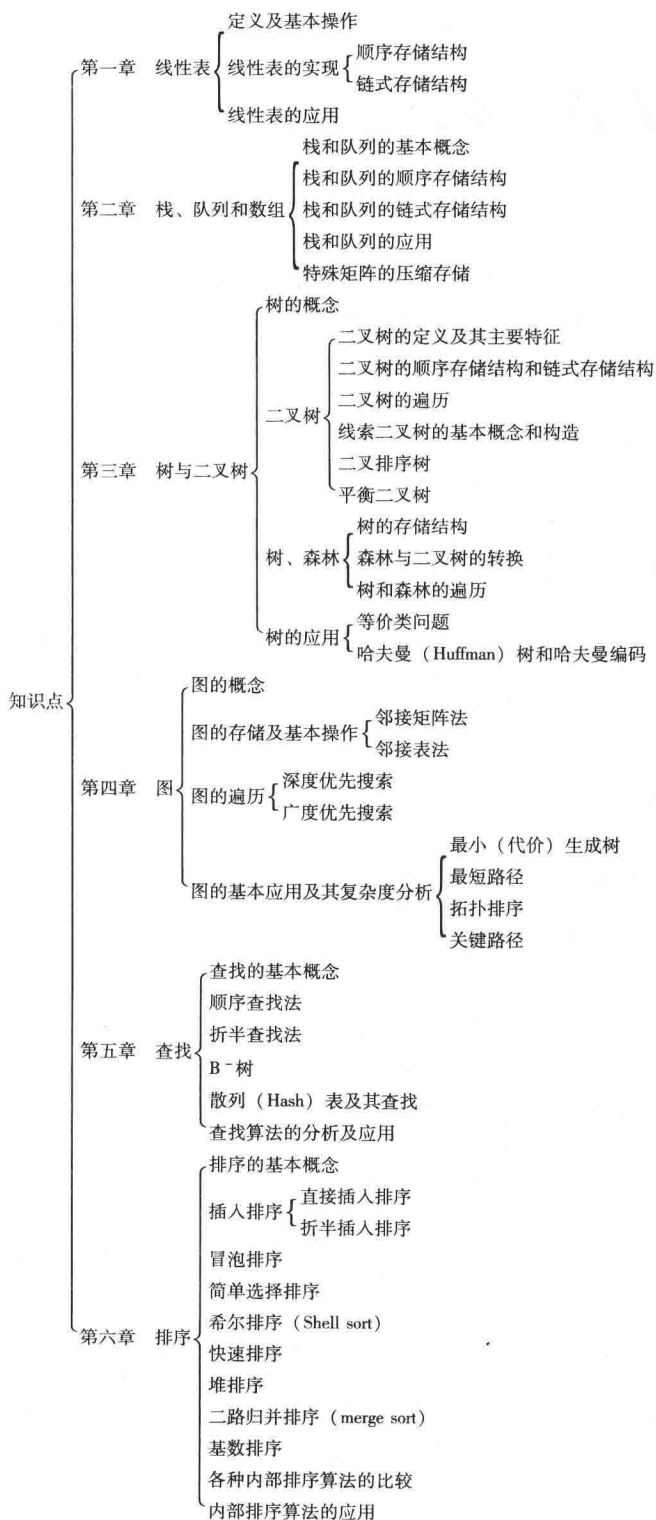
(3) “图”中去掉了对“图的应用的复杂度分析”的要求。这一修改理由和第(2)点类似。

(4) “查找”这一点中,增加了对 B⁻树基本操作的要求,以及 B⁺树的基本概念,考生需要注意 B⁻树和 B⁺树概念的对比。

(5) “查找”这一点中去掉了对“Hash 表查找”的要求,这个修订的理由和第(2)点类似。



知识框架





第一章 线性表



线性表是一种最简单的数据结构。在线性表方面，主要考查线性表的定义和基本操作、线性表的实现。在线性表实现方面，要掌握的是线性表的存储结构，包括顺序存储结构和链式存储结构，特别是链式存储结构，是考查的重点。另外，还要掌握线性表的基本应用。

第一节 线性表的逻辑结构

一、复习要点

线性表的定义和基本操作：线性表的逻辑结构，是指线性表的数据元素间存在着线性关系。主要是指：除第一个及最后一个元素外，每个结点都只有一个前趋和只有一个后继。

线性表是考研的重中之重，近3年的算法设计题都是基于线性表（顺序表或单链表）的。顺序表的运算本质上是对数组进行操作，因此，可能会与排序、查找的内容结合出题。单链表由于结构简单、应用灵活、难度适中，是数据结构的重要考点，考生要重点掌握。

二、考点精讲

1. 线性表的定义

线性结构：数据结构中线性结构指的是数据元素之间存在着“一对一”的线性关系的数据结构。例如， $(a_1, a_2, a_3, \dots, a_n)$ ， a_1 为第一个元素， a_n 为最后一个元素，此集合即为一个线性结构的集合。常用的线性结构有线性表、栈、队列、双队列、数组和串。

线性表：线性表（linear list）是具有相同数据类型的 n （ $n \geq 0$ ）个数据元素 $a_1, a_2, \dots, a_n$ 组成的有限序列。其中 n 为数据元素的个数或线性表的长度。一个数据元素可以由若干个数据项组成。数据元素称为记录，含有大量记录的线性表又称为文件。

◇ 当 $n=0$ 时称为空表， $n>0$ 时称为非空表。

◇ 数据元素 a_i （ $1 \leq i \leq n$ ）是线性表中第 i 个数据元素，它是一个抽象的符号，其数据类型可以根据具体情况而定， i 称为数据元素 a_i 在线性表中的位序。

2. 线性结构基本特征

- (1) 存在唯一的一个称为“第一个”的元素。
- (2) 存在唯一的一个称为“最后一个”的元素。
- (3) 除第一个元素外，集合中的每个元素均只有一个直接前驱。
- (4) 除最后一个元素外，集合中的每个元素均只有一个直接后继。

最后两个特征体现了线性表中元素之间的逻辑关系。

理解线性表的要点如下：

① 表中元素具有逻辑上的顺序性，在序列中各元素排列有其先后次序。各个数据元素在线性表中的逻辑位置只取决于其序号。

② 表中元素个数有限。

- ③ 表中元素都是数据元素。就是说，每一个表元素都是不可再分的原子数据。
- ④ 表中元素的数据类型都相同。这意味着每一个表元素占有相同数量的存储空间。
- ⑤ 表中元素具有抽象性。就是说，仅讨论表元素之间的逻辑关系，不考虑元素究竟表示什么内容。

三、典型例题

【例题】在线性表中的每一个表元素都是数据对象，它们是不可再分的_____。

- A. 数据项 B. 数据记录 C. 数据元素 D. 数据字段

【答案】C

【解析】线性表是 n ($n \geq 0$) 个数据元素的有限序列。数据记录、数据字段是数据库文件组织中的术语。数据项相当于数据元素中的属性。

第二节 线性表的顺序存储结构

一、复习要点

- (1) 线性表的顺序存储结构，靠元素存储的先后位置反映数据元素的逻辑关系。
- (2) 在具体语言环境下有两种不同实现：表空间的静态分配和动态分配。
- (3) 用向量（一维数组）表示，即给定下标可以存取相应元素，属于随机存取的存储结构。
- (4) 线性表的顺序存储结构实现插入、删除、定位等运算的算法。

二、考点精讲

1. 顺序表的定义

定义：顺序表就是把线性表中的所有元素按照其逻辑顺序，依次存储到存储器中从指定存储位置开始的一块连续的存储空间中。需注意的是，该连续存储空间所能容纳的数据元素个数不得小于顺序表指定长度。

特点：在顺序表中，逻辑关系相邻的两个元素在物理位置上也相邻。顺序表用物理上的相邻实现数据元素之间的逻辑相邻关系，即逻辑顺序与物理顺序一致。

数据位置：假设线性表中元素为 $(a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$ ，设第一个元素 a_1 的内存地址为 $LOC(a_1)$ ，而每个元素在计算机内占 t 个存储单元，则第 i 个元素 a_i 的首地址 $LOC(a_i)$ 为

$$LOC(a_i) = LOC(a_1) + (i-1) \times t \quad (\text{其中}, 1 \leq i \leq n)$$

这样只要知道顺序表首地址和每个数据元素所占地址单元的个数就可求出第 i 个数据元素的地址。

2. 顺序表的基本操作

- (1) 顺序表的初始化。将线性表置为空表，代码如下：

```
Void InitList(SeqList &L){
    L.elem = new ElemType[ListSize];
    If(L.elem == NULL){
        Printf("存储分配失败!\n");
        Exit(1);
    }
    L.length = 0; L.maxSize = ListSize;
}
```

顺序表的初始化即构造一个空表，这对表是一个加工型的运算，因此，将 L 设为引用参数。首先

动态分配存储空间, 然后, 将 length 置为 0, 表示表中没有数据元素。

(2) 顺序表中数据元素的插入。

线性表的插入是指在表的第 i (取值范围: $1 \leq i \leq n+1$) 个位置上插入一个值为 item 的新元素, 插入后使原表长为 n 的表 $(a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n)$ 成为表长为 $n+1$ 的表 $(a_1, a_2, \dots, a_{i-1}, \text{item}, a_i, a_{i+1}, \dots, a_n)$ 。

插入一个元素的基本步骤如下:

初始表, 如图 1-1-1 所示。

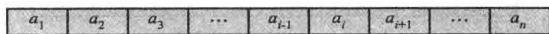


图 1-1-1 待插入顺序表

1) 将 $a_i \sim a_n$ 顺序向下移动, 为新元素让出位置, 如图 1-1-2 所示。



图 1-1-2 在顺序表中插入空位置

2) 将 item 置入空出的第 i 个位置, 如图 1-1-3 所示。



图 1-1-3 在顺序表中插入新元素

3) 修改表长。

用算法描述此过程如下:

```

Bool insert(SeqList &L, ElemType x, int i) {
    // 在表中第 i (1 ≤ i ≤ n+1) 个位置插入新元素 x
    if (i < 1 || i > L.length + 1) return 0;
    if (L.length == ListSize) return 0;
    // 参数不合理或表已满, 插入不成功
    For (int j = L.length - 1; j >= i - 1; j--)
        L.elem[j + 1] = L.elem[j];
    L.elem[i - 1] = x;      // 实际插在第 i - 1 个位置
    L.length ++;
    Return 1;              // 插入成功
}
    
```

算法性能分析:

顺序表上的插入运算, 时间主要消耗在数据的移动上, 在第 i 个位置上插入 item, 从 a_i 到 a_n 都要向后移动一个位置, 共需要移动 $n-i+1$ 个元素。在顺序表中, 插入的位置可以是 1 到 n 中的任何一个, 即在任意一个位置上插入一个元素的概率为 $p_i = 1/n$, 在等概率的情况下, 任意位置上插入一个元素, 需要移动的元素次数为 $(1+2+3+\dots+n)/n = n(n+1)/2n = (n+1)/2$, 即在顺序表中进行插入运算的时间复杂度为 $O(n)$ 。

(3) 顺序表中数据元素的删除。

线性表的删除是指在表的第 i (取值范围为 $1 \leq i \leq n+1$) 个位置上的元素删除, 删除后使原表长为 n 的表 $(a_1, a_2, \dots, a_{i-1}, \dots, a_n)$, 成为表长为 $n-1$ 的表, 即 $(a_1, a_2, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$ 。

在顺序表中删除一个元素需要经过以下步骤:

初始表, 如图 1-1-4 所示。

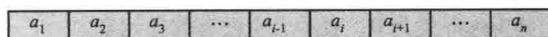


图 1-1-4 待删除顺序表

1) 将第 i 个位置上的元素删除, 如图 1-1-5 所示。

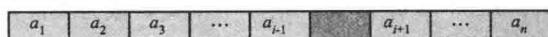


图 1-1-5 删除指定位置元素

2) 将 $a_{i+1} \sim a_n$ 顺序向前移动, 如图 1-1-6 所示。

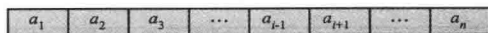


图 1-1-6 删除空闲位置

3) 修改表长。

用算法描述此过程如下:

```
boolDelete(SeqList&L,int i,eLEmTYPE&X){
//在表中删除第 i 个元素,通过 x 返回其值
    If(I>0&&i<=L.length){
        X = L.elem[i-1]; L.length--;
        For(int j=i-1;j<=L.length;j++)
            L.elem[j]=L.elem[j+1];
        return 1;    //成功删除
    }
    return 0;    //表中没有 x
}
```

算法性能分析:

顺序表上的删除运算, 时间主要消耗在数据的移动上。删除第 i 个位置上的元素, 从 a_{i+1} 到 a_n 都要向前移动一个位置, 共需要移动 $n-i$ 个元素。在顺序表中, 删除元素的位置可以从 1 到 n 中的任何一个, 即在任意一个位置上删除元素的概率为 $p_i = \frac{1}{n}$, 在等概率的情况下, 任意位置上删除一个元素, 需要移动的元素次数为 $\frac{1+2+3+\cdots+n}{n} = \frac{n(n+1)}{2n} = \frac{n+1}{2}$, 即在顺序表中进行删除运算的时间复杂度为 $O(n)$ 。

(4) 顺序表中按值查找数据。

在顺序表中按值查找数据是指在线性表中查找与给定的 item 相等的数据元素。最简单的方法就是从第一个元素 a_1 起依次和 item 相比较, 直到找到一个与 item 相等的数据元素。

用算法描述此过程如下:

```
int Find(SeqList &L,ElemType x){
    int i=1;ElemType *p;
    for(p=L.elem;p;p++)    //p++ 将指针 p 进到 L.elem 的下一个元素位置
        if(*p!=x)i++;
        else break;
    if(i<=L.length)return i;    // *p 取 p 所指元素的值
    else return 0;
}
```

算法的性能分析:

按值查找的主要方法是比较, 显然比较的次数与被比较值在表中的位置有关, 也与表长有关。按值查找的评价比较次数为 n , 时间性能为 $O(n)$ 。

(5) 顺序表效率分析。

1) 时间效率分析。算法时间主要耗费在移动元素的操作上, 计算时间复杂度的基本操作:

$$T(n) = O(\text{移动元素次数})$$

移动元素的个数取决于插入或删除元素的位置。

2) 线性表顺序纯粹结构特点: 逻辑关系上相邻的两个元素在物理存储位置上也相邻。

优点: 可以随机存取表中任一元素。

缺点: 在插入、删除某一元素时, 需要移动大量元素。

三、典型例题

【例 1】顺序表是线性表的_____存储表示。

A. 有序

B. 连续

C. 数组

D. 顺序存取

【答案】C

【解析】顺序表是线性表的数组存储表示, 也称为线性表的顺序存储结构。注意, 顺序存取是一种读写方式, 不是存储方式, 有别于顺序存储。

【例 2】下面的叙述, 正确的是_____。

A. 线性表在链式存储时, 查找第 i 个元素的时间与 i 的值无关

B. 线性表在链式存储时, 查找第 i 个元素的时间与 i 的值成反比

C. 线性表在顺序存储时, 查找第 i 个元素的时间与 i 的值成正比

D. 线性表在顺序存储时, 查找第 i 个元素的时间与 i 的值无关

【答案】D

【解析】顺序存储的主要优点: 可以随机存取表中任一元素, 因此, 查找第 i 个元素的时间与 i 的值无关。而链式存储结构中只能按顺序查找元素, 因此, 查找第 i 个元素的时间与 i 的值成正比。

【例 3】通常查找线性表数据元素的方法有(1)和(2)两种方法, 其中(1)是一种只适合于顺序存储结构但(3)的方法; 而(2)是一种对顺序和链式存储结构均可使用的方法。

(1) A. 顺序查找

B. 循环查找

C. 条件查找

D. 折半查找

(2) A. 顺序查找

B. 随机查找

C. 折半查找

D. 分开查找

(3) A. 效率较低的线性查找

B. 效率较低的非线性查找

C. 效率较高的非线性查找

D. 效率较高的线性查找

【答案】(1) D (2) A (3) C

【解析】在线性表中查找指定元素使用顺序查找法和折半查找法。顺序查找法属于线性查找, 效率较低, 但它适用于用顺序方式或用链接方式存储的线性表; 折半查找法仅适用于已排序的顺序存储线性表, 每次根据查找值的大小将查找区间缩小一半继续查找, 因此它不是线性查找, 它比顺序查找的效率高一一些。

第三节 线性表的链式存储结构

一、复习要点

(1) 线性表的链式存储结构, 靠指针来反映数据元素的逻辑关系。

(2) 链表的存取需要从头指针开始，顺链而行，不属于随机存取结构。

(3) 几种常用链表的特点和相关算法设计：单链表、单循环链表、双向链表、双向循环链表的生成、检索、插入、删除、遍历、分解和归并等操作。

(4) 从时间复杂度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及其各自使用的场合。

二、考点精讲

(一) 单链表

1. 单链表的基本概念

(1) 定义。

链表是通过一组任意的存储单元来存储线性表中的数据元素。

单链表的结点结构如图 1-1-7 所示。



图 1-1-7 单链表结构

每个结点除了存储数据元素本身信息的数据域外，只含有一个指针域用来存放其直接后继数据元素的存储地址。线性表通过每个结点的指针域拉成了一个“链”，称之为链表。因为每个结点中只有一个指向后继的指针，所以称其为单链表。

通常用“头指针”来标识一个单链表，如单链表 L、单链表 H 等，是指某链表的第一个结点的地址放在了指针变量 L、H 中，头指针为“NULL”则标识一个空表。

(2) 逻辑结构，如图 1-1-8 所示。

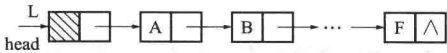


图 1-1-8 单链表逻辑结构

(3) 物理结构，如图 1-1-9 所示。

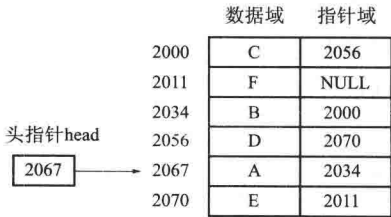


图 1-1-9 单链表物理结构

(4) 单链表存储结构 C 语言描述如下：

```
typedef int DataType;
typedef struct node {
    DataType data;
    struct node *link;
} LinkNode, *LinkList;
```

此时，使用一个指向链表结点的指针 hpt 标识链表的表头：

```
LinkNode *hpt 或 linkList hpt
```