

丛书主编 沈 晖 李立新 王晓敏 王 群
本册主编 吴再陵 高建君

青少年信息学 奥林匹克竞赛 实战辅导丛书

数据结构
及其应用



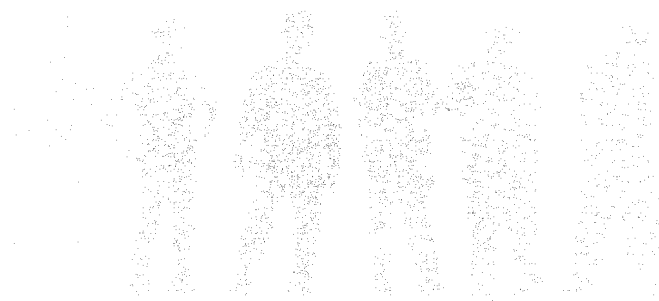
东南大学出版社

青少年信息学奥林匹克竞赛实战辅导丛书

数据结构及其应用

丛书主编 沈 军 李立新 王晓敏

本册主编 王 静 吴再陵 高建君 王 进



东南大学出版社

· 南京 ·

内 容 提 要

本书是面向青少年信息学奥林匹克竞赛活动训练的教材。本书共分为6章,分别是概述(数据结构基础)、线性结构及其应用、线性结构的深入应用、层次结构(树)及其应用、网状结构(图)及其应用、数据结构深入应用。本书以最主要的三种数据组织的逻辑结构(线性、层次、网状)为主线,将数据结构知识恰当地应用到实际问题的解决中,详细分析了每一种逻辑结构的不同的存储方式以及基于不同存储方式建立的对问题求解的算法。每章均配有适量习题。

本书特色是描述简单、易懂、易学,有机地将实际问题求解、竞赛问题求解与数据结构紧密结合,使读者感悟到数据结构与算法之间的关系,体会数据结构的具体应用之奥妙。

本书也可以作为大学生学习数据结构的配套辅导教材。

图书在版编目(CIP)数据

数据结构及其应用/王静等主编. —南京:东南大学出版社, 2009. 3

(青少年信息学奥林匹克竞赛实践辅导丛书 / 沈军, 李立新, 王晓敏主编)

ISBN 978-7-5641-1609-5

I. 数… II. 王… III. 数学结构—教材 IV. TP311.12

中国版本图书馆 CIP 数据核字(2009) 第 032058 号

数据结构及其应用

责任编辑 张 煦
出 版 人 江 汉
出版发行 东南大学出版社
社 址 南京市四牌楼2号(210096)
经 销 江苏省新华书店
印 刷 溧阳市晨明印刷有限公司
版 次 2009年3月第1版 2009年3月第1次印刷
开 本 787 mm × 1092 mm 1/16
印 张 20.75
字 数 479千字
印 数 1—5000册
书 号 ISBN 978-7-5641-1609-5
定 价 34.00元

(凡因印装质量问题,请直接向东大出版社读者服务部调换。电话:025—83792328)

丛 书 序

得益于计算机工具的特殊结构,以计算机技术为核心的信息技术现在已在整个社会发展中起到了极其重要的作用。同时,由于信息技术的本质在于不断创新,因而人们将21世纪称为信息世纪。根据人类生理特征,青少年时期正处于思维活跃、充满各种幻想的黄金年代,孕育着创新的种子和潜能。长期的实践活动告诉我们,青少年信息学奥林匹克竞赛可以让广大的青少年淋漓尽致地展现其思维的火花,享受创新带来的美感。因此,该项活动得到了全国各地广大青少年朋友的喜爱,越来越多的青少年朋友怀着浓厚的兴趣加入到这项活动中来。

从本质上看,计算机学科是一种思维学科,正确的思维训练可以播种持续创新的优良种子。相对于其他学科的竞赛,信息学竞赛覆盖知识面更为宽广,涉及了数学、数据结构、算法、计算几何、人工智能等相关的专业知识。如何在短时间内有效地掌握这些知识的主体,并能灵活地应用其解决实际问题,显然是一个值得认真思考的问题。

知识学习与知识应用基于两种不同的思维策略,尽管这两种策略的统一本质上依赖于选手自身的领悟,但是如何建立两种策略之间的桥梁、快速地促进选手自身的领悟,显然是教材以及由其延伸的教学设计与实施过程所应考虑的因素。竞赛训练有别于常规的教学,要在一定的时间内得到良好的效果,需要有一定的技术方法,而不应拘泥于规范。从学习的本质看,各种显性知识的学习是相对容易的。或者说,只要时间允许,总是可以消化和理解的。然而,隐性知识的学习和掌握却是较难的。由于隐性知识的学习对竞赛和能力的提高起到决定性的作用,因此,仅仅依靠选手自身的感悟,而不能从隐性知识的层面重新组织知识体系,有目的地辅助选手自身的主动建构,显然是不能提高竞赛能力的。基于上述认识,结合多年来开展青少年信息学竞赛活动的经验,我们组织了一批有长期一线教学经验的教练员和专家、教授,编写出版了这套《青少年信息学奥林匹克竞赛实战辅导丛书》。

丛书的主要特点如下:

1. 兼顾广大青少年课外学习时间短暂与知识内容较多的矛盾,考虑我国青少年信息学竞赛的特点和安排,丛书分为四个层次,分别面向日常常规训练、数据结构与数学知识强化(包括基本数据结构与数学知识及应用、高级数据结构及应用)、重点专题解析和典型试题解析,既考虑知识体系的系统性及连续训练



的特点,又考虑各个层次选手独立训练的需要。

2. 区别于常规的教学模式,丛书中每册的体系设计以实战需要为核心主线,突出重点,整个体系从逻辑上构成符合某种知识体系学习规律的系统化结构。

3. 围绕实战辅导需求,在解析知识和知识应用关系所蕴涵的递归思维策略的基础上,重构知识点关系,采用抛锚式和支架式并重的教学思路,突出并强化知识和知识应用两者之间的联系。

4. 在显性知识及其关系基础上,强调知识应用模式及其建构的学习方法的教学,注重学习思维和能力的训练,实现知识应用能力和竞赛能力的提高,强化从程序设计及应用的角度来进行训练的特点。

5. 整套丛书的设计,不仅注重竞赛实战的需要,还考虑选手未来的发展,强调计算机程序设计正确思维的训练和培养,以不断建立持续创新的源泉。

值此邓小平同志“计算机的普及要从娃娃抓起”重要讲话发表 25 周年之际,我们期望以此奉献给广大读者朋友一套立意新、选材精、内容丰富的青少年信息学奥赛读本。

本套丛书的编写与出版,得到了东南大学出版社的大力支持,在此表示衷心的感谢!

沈军 李立新 王晓敏

2008 年 12 月

前 言

本教材遵循青少年信息学奥林匹克竞赛大纲的要求,深入浅出地介绍了数据结构的基本知识、数据结构在程序设计中的作用以及数据结构与算法之间的关系。本教材将数据结构知识与算法设计有机结合,使读者了解数据结构在算法设计中的作用。了解到同一个问题若采用的数据结构不同,其对应的算法实现也不相同;反之一个算法实现也可以采取不同的数据结构。

教材所涉及的内容及知识体系既考虑到中学生现有的知识水平和能力,同时又兼顾到他们在原有知识结构基础上的提高、拓展与创新。并且,将原本高等学校的教学内容,以初等教学的方式呈现在中学生面前,使学生易读、易学、易懂,也有利于教师开展培训工作。

由于我们的水平有限,加之时间比较紧,错误和不足之处在所难免,敬请专家和广大读者批评指正。

编 者

2009 年 2 月

目 录

第 1 章 概 述	1
1.1 相关概念	1
1.1.1 数据	1
1.1.2 数据元素	1
1.1.3 数据类型	1
1.1.4 数据结构	2
1.2 算法	6
1.2.1 算法概念及算法特性	6
1.2.2 算法的描述	8
1.2.3 算法的评价	12
1.3 数据结构与算法	14
习题 1	15
第 2 章 线性结构及其应用	17
2.1 线性表的概念及基本操作	17
2.1.1 线性表的概念	17
2.1.2 线性表的基本操作	18
2.2 线性表的存储结构	18
2.2.1 顺序存储结构	18
2.2.2 链接存储结构	19
2.3 线性表基本操作的实现	22
2.3.1 顺序存储线性表基本操作的实现	22
2.3.2 单链表基本操作的实现	24
2.3.3 双向链表基本操作的实现	28
2.3.4 循环链表基本操作的实现	30
2.4 线性表的应用	32
2.5 特殊线性结构——栈及其应用	38
2.5.1 栈及其基本操作	38
2.5.2 栈的存储方式	38



2.5.3 栈基本操作的实现	40
2.5.4 栈的应用	43
2.6 特殊线性结构——队列及其应用	56
2.6.1 队列及其基本操作	56
2.6.2 队列的存储方式	57
2.6.3 队列基本操作的实现	58
2.6.4 循环队列及其基本操作的实现	61
2.6.5 队列的应用	63
习题2	72
第3章 线性结构的深入应用	83
3.1 高精度运算	83
3.1.1 基本算法	83
3.1.2 应用实例	85
3.1.3 拓展	98
3.2 排序	98
3.2.1 简单排序算法	99
3.2.2 算法的改进	101
3.2.3 应用实例	105
3.3 查找	109
3.3.1 顺序表的查找	109
3.3.2 二分查找	110
3.3.3 索引查找	112
3.3.4 应用实例	112
3.4 散列查找	114
3.4.1 散列表的概念	114
3.4.2 散列函数的构造	115
3.4.3 处理冲突的方法	116
3.4.4 应用实例	120
3.5 分治	126
3.5.1 分治算法解决问题模式	126
3.5.2 应用实例	126
3.6 递推	133
3.6.1 递推算法	133
3.6.2 常见递推关系	140

3.6.3 应用实例	141
3.7 动态规划初探	145
3.7.1 动态规划的定义	145
3.7.2 动态规划的基本概念	148
3.7.3 应用实例	150
习题3	154
第4章 层次结构(树)及其应用	162
4.1 从线性结构到层次结构——广义表及其操作	162
4.1.1 广义表概念及存储结构	162
4.1.2 广义表的建立与输出	164
4.1.3 广义表的应用	165
4.2 树的基本概念	166
4.2.1 树的定义	166
4.2.2 树的表示方法	167
4.2.3 树的基本术语	168
4.3 二叉树的基本知识	168
4.3.1 二叉树基本概念	168
4.3.2 二叉树的性质	168
4.3.3 二叉树的存储结构	170
4.3.4 二叉树的建立算法	171
4.3.5 二叉树的基本运算	172
4.4 二叉树的应用	176
4.5 特殊二叉树及其应用	182
4.5.1 二叉排序树	182
4.5.2 哈夫曼树	184
4.5.3 哈夫曼编码	187
4.6 层次结构的综合应用	188
习题4	194
第5章 网状结构(图)及其应用	200
5.1 网状结构(图)的基本知识	200
5.1.1 图的基本概念	201
5.1.2 图的连通性	203
5.2 图的存储结构	204
5.2.1 邻接矩阵	204



5.2.2 邻接表	205
5.2.3 边集数组	208
5.2.4 邻接压缩表	209
5.2.5 几种存储结构比较	209
5.3 图的遍历	210
5.3.1 图的深度优先遍历	210
5.3.2 图的广度优先遍历	211
5.3.3 应用实例	213
5.4 图的应用	221
5.4.1 求图的某个通路	221
5.4.2 求图的最小生成树	236
5.4.3 求图的最短路径	240
5.4.4 图的拓扑排序及关键路径	246
习题 5	251
第 6 章 数据结构深入应用	257
6.1 概述	257
6.2 从数据结构与算法的关系优化算法	257
6.2.1 数学建模与算法优化	257
6.2.2 时空优化与搜索算法	265
6.3 数据结构与动态规划	277
6.3.1 线性结构与动态规划	277
6.3.2 树型结构与动态规划	289
6.4 综合应用举例	297
6.5 总结	313
习题 6	313
参考文献	320

第1章 概述

【本章学习要点】

- (1) 了解数据、数据结构、算法等基本概念的含义。
- (2) 掌握数据逻辑结构的三种常用类型及表示方法,掌握数据物理结构的常用类型及含义。
- (3) 掌握算法的基本特点及算法的描述方法,并具备基本的算法评价能力。

伴随着计算机的发展,计算机的应用领域从最初的科学计算逐步发展到人类活动的各个领域。现在计算机处理的对象不仅是简单的数值或字符,还有不同结构的各种数据。因此,要设计一个比较好的程序,通过计算机工具处理问题,除了掌握计算机语言外,还需要研究各种数据的特性和数据之间存在的关系,建立方便有效的数据结构,合理组织好用于处理的数据对象。针对同一个问题的求解,若采用的数据结构不同,则建立在其之上的处理问题的算法也不完全相同,处理问题的效率也大相径庭。著名计算机科学家 N. Wirth 指出:算法+数据结构=程序。可见,数据结构在计算机问题处理中具有重要地位,认识和理解各种数据结构成为计算机程序设计的基础。

1.1 相关概念

1.1.1 数据

数据(data)是对客观事物的符号表示。在计算机领域,数据是指所有能输入到计算机中,并能被计算机存储、处理和输出的一切信息,如文字、图形、图像、声音和视频等。

1.1.2 数据元素

数据元素(data element)是数据整体中相对独立的单位,如对于字符串来说,每个字符就是它的数据元素;对于一个数组来说,每个下标变量就是它的数据元素;对于文件来说,每个记录就是它的数据元素;而对于记录中的若干数据项(即数据元素)来说,一个记录又是一个数据。

可见,数据和数据元素是相对而言的,可以不加以严格区分。

1.1.3 数据类型

数据类型是指数据的不同种类,如整数、实数、字符型数、布尔型数等。数据类型的不



同,将引起数据存储时所需要的存储空间大小的不同。不同的数据类型,其数据的运算也不同。因此,在程序设计时,需要根据不同的数据类型,进行相应存储变量的说明,以定义这种类型数据在计算机内存中应该留下的存储空间。

1.1.4 数据结构

数据结构是指数据及数据之间关系的集合。数据及其关系体现在两个层面,一个是逻辑结构,另一个是物理结构。前者用于描述和研究数据及其关系的一些逻辑特性,它不考虑在计算机中的具体实现。后者用于描述和研究数据及其关系的一些物理特性,它需要考虑具体应用中如何实现数据及其关系的存贮。两种结构之间既有区别又有联系。

(1) 数据的逻辑结构

数据元素不是孤立的,数据元素之间存在着一定的逻辑关系,如哪个元素是第一个元素、哪个元素是第二个元素、哪些元素在给定的元素之前或之后等等。数据的逻辑结构一般用二元组描述:

$$DS=(D,R)$$

DS 代表一种数据结构;

D 代表数据元素的集合, $D=\{k_i | 1 \leq i \leq n, n \geq 1\}$, k_i 表示第 i 个数据元素, n 为 D 中数据元素的个数;

R 代表 D 上的二元关系的集合, $R=\{r_j | 0 \leq j \leq m, m \geq 0\}$, r_j 表示第 j 个二元关系, m 为 D 上关系的个数。

通常我们所讲的数据结构是指逻辑结构。常见的数据逻辑结构有:集合结构、线性结构、树型结构(平面层次结构)以及图状结构(平面网状结构)四种。

集合结构:结构中的数据元素之间除了“同属于一个集合”的关系外,别无其他关系。这和数学中的集合概念是一致的。数据结构的直观表示如图 1-1 所示。

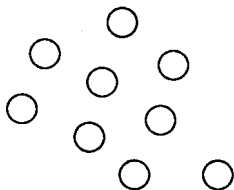


图 1-1 集合结构示例

线性结构:数据结构中的数据元素关系呈现直线状,除了第一个数据元素只有唯一一个后继元素、最后一个数据元素只有唯一一个前驱元素外,其他数据元素都只有唯一的一个后继元素和唯一的一个前驱元素。

【例 1-1-1】10 个学生记录按学号排列

$D=\{S1, S2, S3, S4, S5, S6, S7, S8, S9, S10\}$

$R=\{(S1, S2), (S2, S3), (S3, S4), (S4, S5), (S5, S6), (S6, S7), (S7, S8), (S8, S9), (S9, S10)\}$

数据结构的直观表示如图 1-2 所示。

S1——S2——S3——S4——S5——S6——S7——S8——S9——S10

图 1-2 线性结构示例

特点:数据元素之间的关系是 1:1 的联系(除了第一个数据元素和最后一个数据元素)。

层次结构:数据结构中的数据元素关系呈现层次状,除了第一个数据元素没有前驱数据元素外,其他数据元素都只有唯一的一个前驱数据元素;所有数据元素可以有零个或多个后继数据元素。

【例 1-1-2】一个行政部门的组织机构

$D = \{\text{校长 } P, \text{副校长 } AP1, \text{副校长 } AP2, \text{副校长 } AP3, \text{学生处处长 } SO, \text{教务处处长 } EO, \text{总务处处长 } CO, \text{财务处处长 } AO, \text{学生处员工 } S1, \text{教务处员工 } J1, \text{教务处员工 } J2, \text{总务处员工 } Z1, \text{总务处员工 } Z2, \text{财务处员工 } A1, \text{财务处员工 } A2, \text{财务处员工 } A3\}$

$R = \{(P, AP1), (P, AP2), (P, AP3), (AP1, SO), (AP2, EO), (AP3, CO), (AP3, AO), (SO, S1), (EO, J1), (EO, J2), (CO, Z1), (CO, Z2), (AO, A1), (AO, A2), (AO, A3)\}$

数据结构的直观表示如图 1-3 所示。

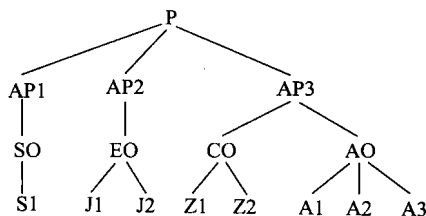


图 1-3 层次结构示例

特点:数据元素之间的关系是 1:N 的关系。

网状结构:数据结构中的数据元素关系呈现网状结构,每个数据元素都可有零个或多个前驱数据元素和零个或多个后继数据元素。也就是说,任意两个数据元素之间都可以有关系或没有关系。

【例 1-1-3】人与人之间互相认识的关系

$D = \{P1, P2, P3, P4, P5, P6, P7\}$

$R = \{(P1, P2), (P1, P3), (P2, P3), (P2, P4), (P2, P5), (P4, P5), (P3, P5), (P4, P6), (P5, P6), (P5, P7), (P6, P7)\}$

数据结构的直观表示如图 1-4 所示。

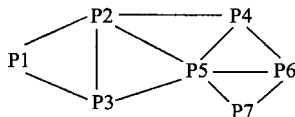


图 1-4 网状结构示例

特点:数据元素之间的关系是 M:N 的关系。

为了区别于线性结构,我们将层次结构和网状结构统称为非线性结构。

(2) 数据的物理结构

数据物理结构是指相关数据在计算机中的具体存储方式。常用的存储方式有顺序、链



接、索引、散列等。

顺序存储结构:把逻辑上相邻的数据元素存储在相邻的存储单元中。顺序存储结构一般用数组来描述和实现,程序运行时数组占用内存中一片连续的存储空间,其存储空间的地址是连续的。因此,这样的存储结构,便于按顺序访问存储空间。用户只要输入数组下标,计算机便可以按顺序计算出地址并访问到该存储单元的内容。但是,顺序存储结构的缺点是,当需要插入或删除一个数据时,将会引起内存中数据的大量移动。

【例 1-1-4】建立线性表

按照学号的顺序,建立 10 个学生的数学成绩档案线性表,档案中包括姓名和成绩。

定义数据元素的类型:

```
type student=record
    name:string[8];
    num:0..100;
end;
```

定义表示内存空间的存储变量:

```
var  a:array[1..20] of student;
    x:integer;
```

建立顺序存储结构的学生数学成绩档案线性表:

```
for  x:=1 to 10 do
begin
    readln(a[x].name);
    readln(a[x].num);
end;
```

此时,建立好的成绩档案线性表如图 1-5 所示。

	name	num
A[1]	li ming	90
A[2]	wang lin	85
A[3]	xue hua	93
	⋮	⋮
A[10]	chen lin	80

图 1-5 成绩档案线性表

通过下列程序片段,可以输出顺序存储结构中的学生数学成绩档案线性表:

```
for x:=1 to 10 do
begin
    write(a[x].name:10);
    writeln(a[x].num:5);
end;
```

当要查找某个学生的成绩时,只要指定该学生记录所对应的序号(数组下标),就能知

道学生姓名及数学成绩。比如,当查找 2 号同学的情况时,则可以用输出语句:

```
writeln(A[2].name:10, A[2].num:10);
```

对于这样的顺序存储结构,如果需要插入一个学生信息或删除一个学生信息,并使其仍然按照学号顺序排列(即仍然维持线性存储结构),将会引起数据的移动,从而降低程序工作的效率。

链式存储结构:顺序存储的变量,是一种静态型变量。程序中一经定义,计算机系统(执行时)就在内存中固定分配相应大小的连续的存贮单元。变量名是存贮单元的地址,而变量的值是存贮单元的内容,该存贮空间自始至终都被占用,直到程序运行结束。如果变量是局部变量,那么在它的作用域内,一经定义也占有一定的存贮单元,直到退出其作用域为止。这样的变量,在程序的执行过程中,不能随时使用随时分配存贮空间,也不能在程序执行的过程中,随时释放这些空间。采用这种静态存贮方式的变量,如字符类型变量、数组类型变量、记录类型变量等,访问方便,查找容易。但它们在处理数据量动态变化的场合有局限性,也就是说,在程序运行过程中,不能按需随意改变存储容量的大小,从而给某些问题的处理带来不便。

为了克服静态存储的缺点,可以采用动态存储结构,如图 1-6 所示是树的链式存储示意图。所谓动态存储结构,是指可以在程序运行过程中,按需动态申请存储空间,并将所获得的存储空间的地址存储到相应的变量中,然后再通过变量之间的关系构成需要的数据存储结构。在存储空间不需要时,也可以立即释放该存储空间并归还给系统,以节省存储空间的消耗。

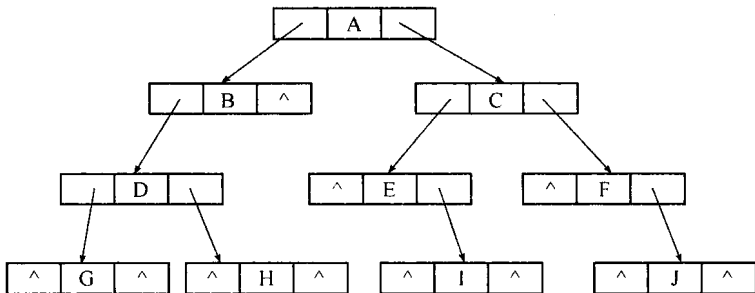


图 1-6 树的链式存储

动态存储结构中,每个数据元素的存储空间不再连续相邻,数据元素之间的逻辑结构关系通过存放地址的变量之间的关系维持和体现。因此,对于线性结构的数据组织,其存储结构相当于一链,每个存储数据元素的存储空间就是链中的一个结点。因此,称这种非连续的物理存储结构为链式存储结构。相应的,称通过链式结构建立的线性表为线性链表。链式存储结构中,数据元素之间的逻辑关系与其实际存储的存储空间之间的关系不再一一对应。

链式存储结构为了维持数据元素之间的逻辑关系,通过扩展数据元素存储的变量结构来实现。也就是说,每个数据元素存储空间由存放数据值本身的变量和一个存放其他数据元素存储空间地址的变量组成。因此,链式存储结构是通过空间换取时间。因为,在链式存储结构中,删除或插入一个数据元素时,只需要通过修改相关数据元素存储空间中存放地址的变量的值来完成,而不需要大量移动数据元素,从而提高程序的执行效率。

索引存储结构:首先把一个线性表(主表)按照一定的函数关系或条件划分成若干个子



表,为每个子表分别建立一个索引项,由所有这些索引项构成主表的一个索引表,于是采用顺序或链接的方式来分别存储索引表和每个子表。图 1-7 是电话号码查询问题及索引存储示意图。

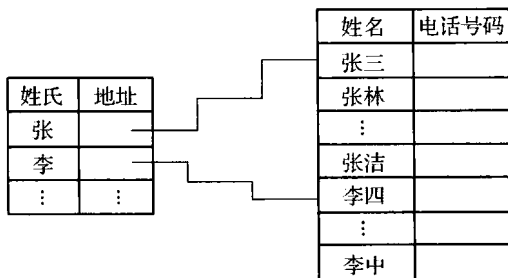


图 1-7 电话号码查询问题的索引存储

散列存储结构:如图 1-8 所示,以线性表中的每个元素的关键字 k 为自变量,通过一种函数 $h(k)$ 计算出函数值,把这个值解释为一块连续存储空间(即数组空间)的单元地址(即下标),将该元素存储到这个单元中。散列存储中使用的函数 $h(k)$,称为散列函数或哈希函数。

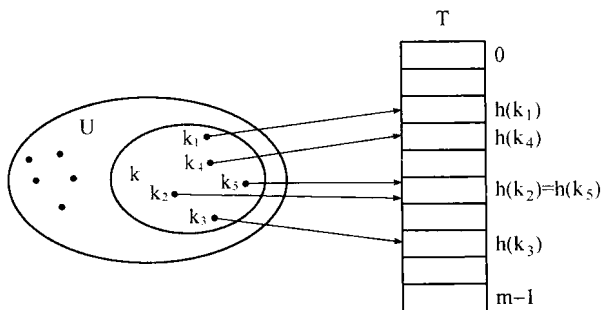


图 1-8 用散列函数 h 将关键字映射到散列表中

1.2 算 法

1.2.1 算法概念及算法特性

(1) 算法的概念

计算机之所以能解决问题,是因为人们事先安排好了计算机解决问题的方法和执行的步骤。为解决一个问题而采取的方法和步骤,称为“算法”。实际上,处理任何问题都需要算法,比如中国象棋有中国象棋的棋谱,国际象棋有国际象棋的棋谱;再比如,邮寄物品有其相应的手续,购买飞机票也有一系列的手续等。

对于同一个问题,往往会有不同的算法。比如:

问题 1: 有 8 个小球,其中 7 个重量相同,仅有一个较重,用天平如何称出那个重的小球?

算法 a:



把8个小球分成4组,依次将每组放在天平上,直到某一组天平不平衡,就可确定重的小球,最多需要称4次。

算法 b:

- ① 从8个小球中任取6个小球,将这6个小球每边3个置于天平上;
- ② 若天平平衡,则表明重的小球在剩余的2个小球中,只需将那2个小球放在天平上再称一次就可找到重的那个小球;
- ③ 若天平不平衡,则从较重的一边的3个球中任取2个球称量,若平衡,则剩下的那个即为要找的那个小球,若不平衡,则重的那边就是要找的小球。

算法 b 只需要称2次即可,比算法 a 优越。

问题 2: 找出1至1000间7的倍数。

算法 a:

- ① 设 $A=0$;
- ② 将 A 不断加1,然后将 A 除以7,若余数为0,则找到了一个7的倍数,将其输出;
- ③ 反复做步骤②,直到 $A=1000$ 结束。

共需完成除法1000次。

算法 b:

- ① 设 $K=1$;
- ② 输出 $K \times 7$ 的值;
- ③ 将 K 增加1,若 $K \times 7$ 的值小于1000,则回到步骤②,否则算法结束。

共需完成乘法142次。

算法 c:

- ① 设 $X=7$;
- ② 输出 X 的值;
- ③ 将 X 的值增加7,若结果没有超过1000,则回到步骤②,否则算法结束。

共需完成加法141次。

上述三个算法中,算法 c 的执行时间最短。

由此可见,算法实质上是针对所处理问题的需要,在数据结构基础上所施加的一种运算。也就是说,一方面算法与问题相关,另一方面算法与采用的数据结构相关。

(2) 算法的特性

一个完整的算法应该具有下列特性:

确定性: 算法的每一步必须是确切定义的,且无二义性。算法只有唯一的一条执行路径,对于相同的输入只能得出相同的输出。

有穷性: 一个算法必须在执行有穷次运算后结束。在所规定的时间和空间中,若不能获得正确结果,其算法也是不能被采用的。

可行性: 算法中的每一步骤必须能用实现算法的工具——可执行指令精确表达,并在有限步骤内完成,否则这种算法也是不会被采纳的。

有0个或多个输入: 算法可以有输入的初始数据,也可以没有给定的初始数据。

有输出: 算法一定能得到问题的解,有一个或多个结果输出,达到求解问题的目的。没有输出结果的算法是没有意义的。