

新编奥林匹克 信息学竞赛指导

——数据结构与算法设计

王建德

南京师范大学出版社

序 摇 言

近几年来,计算机技术飞速发展,计算机应用日益普及,特别是网络提供的各种服务深刻影响着人们日常的工作、学习、交往、娱乐等活动方式。从更深的层次上讲,以计算机技术为核心的信息技术极大地改变了人类的思考方式和知识获取的途径。世界各国紧紧抓住这一机遇,重新调整人才的培养模式,使学生的学习从传统的知识习得型向能力发展型过渡,掌握捕捉、组织和处理信息的能力以及用整体、系统的观念处理复杂问题的方法。应该说,计算机文化的内涵更加丰富了,计算机文化教育的必要性已成为大家的共识。

为了普及计算机知识,活跃中学生课余生活,选拔学有专长的优秀学生,各国基础教育界竞相举办计算机竞赛。例如:我国有全国奥林匹克信息学竞赛分区联赛,全国奥林匹克信息学竞赛(预赛),国际奥林匹克信息学竞赛中国队赛(预赛),国际上有国际奥林匹克信息学竞赛(预赛),等等。这些颇具影响力和权威性的竞赛都是以程序设计为主的。有人问,计算机技术是多方面的,为什么考核内容不选基础教育课程内的文字处理、多媒体和网络知识,而偏偏选中程序设计呢?这是因为:

程序设计与软件工具不同,它要求编程者以某种高级语言为媒介,通过构造算法去解决由现实生活中

抽象出来的各种问题,这些问题非一般软件工具所能解决。如果说计算机应用是“人脑延伸”的话,程序设计即为这种延伸的最高形式。

程序设计对人的能力要求是多方面的。编程者不仅要熟悉计算机语言功能,而且还必须具备:

- (员)扎实的数学基础和算法知识,能够对问题或客观存在的事物及其所要解决的问题产生正确的认识和理解,包括弄清事物属性、行为及其彼此之间的关系;
- (圆)娴熟的编程技术,能够把对问题及其方法的认识描述出来,最终产生一个计算机能够理解和执行的系统实现;
- (猿)相应的实践能力和创造能力,能够独立思考、提出质疑,拓延思路、洞悉规律,创造性地运用知识于不同问题情景。

正因为程序设计能比较客观地反映人的综合素质,因此国际、国内的奥林匹克信息学竞赛都将其作为考核内容。过去是这样,现在是这样,将来恐怕亦不会随计算机技术的发展而改变。

何谓程序设计?有一句至理名言回答了这个问题:“程序设计越数据结构 垣算法”。程序设计一般分两步:第一步是宏观设计,定义数据模型级上的运算步骤。在这一步中不需要明确变量的数据结构,运算带有抽象性质,不含具体细节。第二步是微观设计,为每一个变量定义数据结构,为每一个抽象运算编写函数或过程说明。微观设计是宏观设计的具体实现,它依赖于宏观设计中定义的那些抽象运算。

同样,只有通过微观设计选择数据模型的具体表示形式,才能最终实现宏观设计中定义的抽象运算并确定其效率。由此可见,数据类型与定义在该类型上的运算是彼此依赖、互相制约、融为一体的,它是面向对象的程序设计的基础。本书将“数据结构”和“算法分析”融合成一本教材,正是体现了两者间密不可分的关系。在“数据结构”一篇中,作者由浅入深地介绍了线性表、非线性表的知识 and 应用;在“算法分析”一篇中,作者详尽地介绍了常用的一些算法和解题策略,介绍了算法复杂度和编程技术。这些内容基本涵盖了信息学竞赛的知识结构。本书在系统性、入门性和实用性上颇有特色,始终围绕编程实践,积极创设相关知识的问题情景,讲过程、讲思路、讲方法。理论描述准确,文字通俗流畅,为分析理论展开的比喻、图示和例题循循善诱,引人入胜。书中不仅介绍了竞赛常用的数据结构和算法,而且还根据竞赛需要引进了一些新知识、新成果,这些内容在目前同类大学教材中也尚未介绍过,因此本书既可以作为各省、市信息学竞赛培训活动的教材,亦可以作为大学计算机专业学生查阅数据结构和算法的参考手册。

全书例举的示例是从十年来国际和国内信息学竞赛的试题中精选出来的,很有典型意义。书中并未为其提供直接上机运行的源代码,而是采用比较贴近自然语言的类孕释来描述算法的基本思想和步骤的,这就为读者上机实践留下了空间。“纸上得来终觉浅,绝知此事要躬行”。读者不妨亲手做一做,通过上机获得真知。读者今后面临的实际问题千奇百怪,数据模型千姿百态,求解方法千变万化。究竟采用哪一种算法,设计哪一类数据结构,要求读者

从实际出发 ,因地因时制宜 ,自己想方设法。只有不断通过实践 ,才能渐入本书分析和题解所描述的那种佳境。“阵而后战 ,兵法之常 ,运用之妙 ,存乎一心。”

中国计算机学会常务理事、上海市计算机学会理事长
教育部计算机教学指导委员会副主任
复旦大学首席教授、博士生导师

施伯乐

~~周田田~~愿

目 录

第一篇 数据结构

第一章 顺序存储结构的线性表	(猿)
摇 异质数组	(猿)
摇 异质圆栈	(猿)
摇 异质圆队列	(猿)
摇 异质源串	(猿)
第二章 链式存储结构的线性表	(猿)
摇 异质单链表	(猿)
摇 异质双向循环链表	(猿)
第三章 非线性结构(圆)—树	(猿)
摇 异质树的基本概念及其存储结构	(猿)
摇 异质二叉树	(猿)
摇 异质最优二叉树	(猿)
摇 异质源树的应用举例	(猿)
第四章 非线性结构(圆)—图	(猿)
摇 异质图的基本概念和存储结构	(猿)
摇 异质图的遍历和生成树	(猿)
摇 异质最短路径问题	(猿)
摇 异质源拓扑排序和计算关键路径	(猿)
摇 异质图的应用实例	(猿)

第二篇摇算法设计

第五章摇算法设计的常用策略	(员愿)
摇 异缘员对应的策略	(员愿)
摇 异缘圆分治策略	(员愿)
摇 异缘猿归纳策略	(员猿)
摇 异缘源枚举策略	(圆园)
摇 异缘缘模拟策略	(圆源)
第六章摇搜索隐式图的基本算法	(圆缘)
摇 异缘员回溯法	(圆缘)
摇 异缘圆广度优先搜索	(圆源)
第七章摇动态程序设计	(猿猿)
摇 异缘员动态程序设计方法的基本思路	(猿愿)
摇 异缘圆动态程序设计的应用实例	(猿猿)
第八章摇网络流及其应用	(猿缘)
摇 异缘员基本概念与基本定理	(猿缘)
摇 异缘圆寻求最大流的标号法(标号法 增广路算法)	(猿缘)
摇 异缘猿标号法的应用实例	(猿猿)
摇 异缘源标号法的推广	(猿猿)
第九章摇算法分析与 晕孕问题简介	(源圆)
摇 异缘员算法的正确性和测试	(源愿)
摇 异缘圆算法的简明性	(源圆)
摇 异缘猿算法的时间复杂度	(源缘)
摇 异缘源算法的空间复杂度	(源员)
摇 异缘缘算法的最优性	(源猿)
摇 异缘远晕孕问题简介	(源员)

第一篇 数据结构

早期的程序设计主要偏重于数值计算领域,采用的数据结构相对简单。例如 ~~云~~ 语言仅定义了数组(包括多维数组)和复数两种结构型数据,这两种数据类型足以应付当时多数的科学计算问题。

但是随着现代科技的发展,计算机逐渐应用于数据处理和非数值计算问题,从客观事物中抽象出的数据日益显现出多样化的特征,简单的数据类型已远远不能满足需要,各数据元素之间的复杂联系已经不是普通的数学方程式所能表达的了。在这种背景下,一种专门研究数据之间结构关系的学科——数据结构便应运而生。

数据结构专门研究各种数据的表示、数据的类型以及它们之间关系的集合,其研究范围主要包括各种数据结构的性质,即它们的逻辑结构、物理结构以及施于其上的操作。数据结构的类型繁多,可以从不同的角度来划分:若从数据元素的值在使用时具有不可分割的性质或者是它可以由更基本的成份组成这个角度来划分,数据结构可以分成简单类型和构造类型两大类;如果从数据所占据的内存空间在程序执行期间是否发生变化这个角度来划分,数据结构又可以分成静态结构和动态结构两大类;如果从数据结点后继关系的多少和是否具有层次性的角度划分,数据结构还可以分成线性结构和非线性结构两大类。

简单类型	整型、实型、字符型、布尔型	静态数据类型
构造类型	数组、记录、集合、字符串	
		文件、指针
线性结构	数组、栈、队列、链表、串	
非线性结构	树、图	

通常高级程序设计语言都提供了各种简单类型和静态构造类型的数据结构。例如 *Fortran* 就提供了 5 种类型的定义。这 5 种类型中除了文件和指针属于动态结构的构造类型外,其余 4 种均属于简单类型和静态构造类型。在上表的数据结构中,像数组、栈、串和队列等数据结构属于线性数据结构,而树和图属于非线性数据结构。线性数据结构易于表示各结点之间的联系,其存储方式相对简单;非线性数据结构往往能比较形象地反映各结点之间的层次关系。无论是线性结构或非线性结构,若采用数组方式存储,则属于静态数据类型,若采用指针类型存储,则属于动态数据类型。考虑到篇幅限制和读者大多具备 *Fortran* 语言或 *Pascal* 语言的基础,本书侧重讲解线性结构和非线性结构两种。

数据结构和算法有着密切的联系,简捷有效的算法很大程度上出自于对数据结构的正确选取。奥林匹克信息学竞赛的试题大都属于非数值计算问题,从问题中抽象出的数据多半是结构类型的,因此,对于参与这项活动的学生来说,学好、用好数据结构尤为重要。为此,我们在本书中详尽地介绍了数据结构的有关概念和基本操作,同时辅之以一些实例,围绕编程实际展开讨论,尽可能多给读者一点启示。

第一章 顺序存储结构的线性表

线性表是最常用且比较简单的一种数据结构,它是由有限个数据元素组成的有序集合,每个数据元素有一个数据项或者含多个数据项。例如 26 个英文字母表(a, b, …… z) 是一个线性表,表中每一个数据元素由单个字母组成数据项。又如图 1.1 也是一个线性表,表中含 4 个数据元素,每一个数据元素由 3 个选手该项目的竞赛成绩组成。

学生 项目	学生 1	……	学生 3	……	学生 4	项目总分
项目 1						
……						
项目 2			a_{23}			$\sum_{i=1}^4 a_{2i}$
……						
项目 4						
选手总分	$\sum_{j=1}^4 a_{1j}$		$\sum_{j=1}^4 a_{3j}$			

图 1.1

线性表具有如下结构特征：

均匀性：即同一线性表的各数据元素的数据类型一致且数据项

数相同。

有序性 表中数据元素之间的相对位置是线性的,即存在惟一的“第一个”和“最后一个”数据元素。除第一个和最后一个外,其它元素前面均只有一个数据元素(直接前趋),后面也均只有一个数据元素(直接后继)。

按照表中数据元素的存储方式分为顺序存储结构和链式存储结构两类线性表。顺序存储结构是指用一组地址连续的存储单元依次存储线性表的元素,通常用数组实现;而在链式存储结构的线性表中,逻辑上相邻的两元素,其物理位置不要求相邻,其实现既可采用静态数组,亦可采用动态指针。为了扩大用户空间和更好地体现链式存储结构的便利,实践中大都采用动态链表形式。为此,本书在讲解顺序存储结构的线性表时采用数组类型,在讲解链式存储结构的线性表时采用指针类型。另外,根据存取方式的限制和表元素类型的限制,还存在几种特殊类型的线性表:栈、队列、串,它们一般采用顺序存储结构。

异 源 数 组

我们称有限个同类型数据元素的序列为数组,它是一种定长的线性表。若数组元素不再有分量,该序列叫一维数组,若数据元素为数组,则称该元素为多维数组。例如:

增删

摇粤 葬 糟 · 凿 燥 藻

摇月 葬 糟 · 凿 糟 · 凿 燥 藻

粤为一维数组,表长为(凿 糟 员),元素类型为 葬 糟 藻 月为二维数组,表长为(凿 原 糟)(凿 原 糟),元素类型为 凿 糟 藻 一维数组和

多维数组的元素个数由其类型说明或变量说明静态确定,执行中不能

· 源

增减其内存空间。

一、数组的顺序存储结构

数组的物理实现是一块连续的存储空间,它是按首址(表中第 员个元素的地址)垣位移来访问每一个元素的。设:

葬糟葬蚤) — 粤表中元素 蚤的内存地址(糟 ≤ 蚤 ≤ 葬);

葬糟遭蚤躁) — 月表中元素(蚤躁)的内存地址(糟 ≤ 蚤 ≤ 葬, 糟 ≤ 躁 ≤ 葬);

葬糟葬蚤) 越 葬糟葬糟) 垣 (蚤糟 * 葬) 摇葬 — 葬糟葬糟) 类型的长度。

葬糟 (遭 [蚤 躁]) 越 葬糟遭糟) 垣

$$\frac{((葬原糟) * (蚤糟) 垣 躁糟) * 葬}{位移}$$

葬 — 葬糟葬糟) 类型的长度。

一维数组按照下标递增的顺序访问表中元素:

葬糟 → 葬糟葬] → ... → 葬。

二维数组按照先行后列的顺序访问表中元素:

葬糟糟] → 葬糟糟垣] → ... → 葬糟,葬] → ... → 葬,葬原] → 葬,葬] → ... → 葬,葬原] → 葬,葬]。

在数组中,数据元素的下标间接反映了数据元素的存储地址。而计算机内存是随机存取装置,所以在数组中存取一个数据元素只要通过下标计算它的存储地址就行了,数组中任意一个元素的存取时间都相等。从这个意义上讲,数组的存储结构是一个随机存取的结构。

二、数组在顺序存储结构下的插入与删除

设数组定义为:

葬糟葬垣 越 数组的长度上限;

数)。

设 $曾_{员灶愿}$ 表示编号为 $蚤$ 的选手第 $躁$ 项竞赛的成绩 ($员 \leq 蚤 \leq 灶$, $员 \leq 躁 \leq 愿$)。其它指标如下：

(员) 第 $躁$ 项竞赛的平均分 $葬_{躁} = \frac{1}{灶} \sum_{蚤=1}^{灶} 曾_{蚤躁}$ ($员 \leq 躁 \leq 愿$),

(圆) 选手 $蚤$ 的总分 $葬_{蚤} = \sum_{躁=1}^{愿} 曾_{蚤躁}$ ($员 \leq 蚤 \leq 灶$),

(猿) 选手 $蚤$ 第 $躁$ 项竞赛的位置分

$$赠_{蚤躁} = \begin{cases} 园 & \text{若 } \sum_{蚤=1}^{灶} |曾_{蚤躁} - 葬_{躁}| < 越园 \\ \frac{曾_{蚤躁} - 葬_{躁}}{\sum_{蚤=1}^{灶} |曾_{蚤躁} - 葬_{躁}|} & \text{若 } \sum_{蚤=1}^{灶} |曾_{蚤躁} - 葬_{躁}| \neq 园 \end{cases} \quad (员 \leq 蚤 \leq 灶, 员 \leq 躁 \leq 愿)$$

(源) 选手 $蚤$ 的总位置分 $葬_{蚤} = \sum_{躁=1}^{愿} 赠_{蚤躁}$ ($员 \leq 蚤 \leq 灶$)。

排名规则如下：

(员) 总位置分高的选手名次在前；

(圆) 若两个或两个以上的选手总位置分相同, 则总分高的选手名次在前；

(猿) 若两个或两个以上的选手总位置分和总分均相同, 则编号在前的选手名次在前。

请你为竞赛组委会编一程序, 计算本次全能竞赛的总排名情况。

输入输出 输入文件为 `input.txt` 文件的第 $员$ 行为参赛总人数 $灶$ ($员 \leq 灶 \leq 100$), 从第 $圆$ 行到第 $灶$ 行依次为编号为 $员$ 到编号为 $灶$ 的选手的成绩, 每行有 $愿$ 个 $0 \sim 100$ 之间的整数, 代表该选手的愿项竞赛成绩 $曾_1, 曾_2, \dots, 曾_愿$ 。同一行相邻两个数之间用一个空格符隔开。

输出文件为 `output.txt` 文件有 $灶$ 行, 每行依次为排名第 $员$ 的选手的编号, 排名第 $圆$ 的选手的编号... 排名第 $灶$ 的选手的编号。

样例：