



21st CENTURY
规划教材

面向21世纪高职高专计算机系列规划教材

COURSES FOR VOCATIONAL HIGHER EDUCATION, COMPUTER

数据结构

DATA STRUCTURES

方风波 王巧莲 主 编



科学出版社
www.sciencep.com



面向21世纪高职高专计算机系列规划教材
COURSES FOR VOCATIONAL HIGHER EDUCATION: COMPUTER

数 据 结 构

方风波 王巧莲 主编

黄鹤鸣 副主编

科学出版社

北 京

内 容 简 介

本书为高职高专计算机及相关专业的教材。全书共分10章,分别为绪论、线性表、栈和队列、串、数组和广度表、树、图、排序及查找,上机指导和实验,基本上覆盖了数据结构的所有知识。全书用C语言作为算法描述语言,详细介绍了各种数据结构的逻辑特征、存储表示和有关运算的算法,内容丰富、通俗易懂。为便于巩固教学,各章后都附有大量习题。

本书既可作为高职高专计算机专业的教材,也可以作为计算机自学的教材。

图书在版编目(CIP)数据

数据结构/方风波,王巧莲主编. —北京:科学出版社,2004

(面向21世纪高职高专计算机系列规划教材)

ISBN 7-03-013932-1

I. 数… II. ①方… ②王… III. 数据结构-高等学校:技术学校-教材
IV. TP311.12

中国版本图书馆CIP数据核字(2004)第071158号

责任编辑:李昱顿/责任校对:柏连海

责任印制:吕春珉/封面设计:飞天创意

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

新蕾印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2004年7月第 一 版 开本:787×1092 1/16

2004年7月第一次印刷 印张:11 1/2

印数:1—4 000 字数:255 000

定价:16.00 元

(如有印装质量问题,我社负责调换〈路通〉)

面向 21 世纪高职高专规划教材专家委员会

主 任 李宗尧

副主任 (按姓氏笔画排序)

丁桂芝 叶小明 张和平 林 鹏
黄 藤 谢培苏

委 员 略

信息技术系列教材编委会

主 任 丁桂芝

副主任 (按姓氏笔画排序)

万金保 方风波 徐 红 鲍 泓

委 员 (按姓氏笔画排序)

于晓平	马国光	仁英才	王东红	王正洪
王 玉	王兴宝	王金库	王海春	王爱梅
邓 凯	付百文	史宝会	本柏忠	田 原
申 勇	任益夫	刘成章	刘克敏	刘甫迎
刘经玮	刘海军	刘敏涵	安志远	许殿生
何瑞麟	余少华	吴春英	吴家碚	吴瑞萍
宋士银	宋锦河	张红斌	张环中	张海鹏
张蒲生	张德实	李云程	李文森	李 洛
李德家	杨永生	杨 闯	杨得新	肖石明
肖洪生	陈 愚	周子亮	周云静	胡秀琴
赵从军	赵长旭	赵动庆	郝 梅	唐铸文
徐洪祥	徐晓明	袁德明	郭庚麒	高延武
高爱国	康桂花	戚长政	曹文济	黄小鸥
彭丽英	董振珂	蒋金丹	韩银峰	魏雪英

本书编写人员名单

主 编 方风波 王巧莲

副主编 黄鹤鸣

撰稿人(按姓氏笔画排序)

李 军 李太芳 耿 杰

出版前言

随着世界经济的发展,人们越来越深刻地认识到经济发展需要的人才多元化、多层次的,既需要大批优秀的理论性、研究性的人才,也需要大批应用性人才。然而,我国传统的教育模式主要是培养理论性、研究性的人才。教育界在社会对应用性人才需求的推动下,专门研究了国外应用性人才教育的成功经验,结合国情大力度地改革我国的“高等职业教育”,制定了一系列的方针政策。联合国教科文组织 1997 年公布的教育分类中将这种教育称之为“高等技术与职业教育”,也就是我们通常所说的“高职高专”教育。

我国经济建设需要大批应用性人才,呼唤高职高专教育的崛起和成熟,寄希望于高职高专教育尽快向国家输送高质量的紧缺人才。近几年,高职高专教育发展迅速。目前,各类高职高专学校已占全国高等院校的近 1/2,约有 600 所之多。教育部针对高职高专教育出台的一系列政策和改革方案主要体现在以下几个方面:

- “就业导向”成为高职高专教育的共识。高职高专院校在办学过程中充分考虑市场需求,用“就业导向”的思想制定招生和培养计划。
- 加快“双师型”教师队伍建设。已建立 12 个国家高职高专学生和教师的实训基地。
- 对学生实行“双认证”教育。学历文凭和职业资格“双认证”教育是高职高专教育特色之一。
- 高职高专教育以 2 年学制为主。从学制入手,加快高职高专教学方向的改革,充分办出高职高专教育特色,尽快完成紧缺人才的培养。
- 开展精品专业和精品教材建设。已建立科学的高职高专教育评估体系和评估专家队伍,指导、敦促不同层次、不同类型的学校办出一流的教育。

在教育部关于“高职高专”教育思想和方针指导下,科学出版社积极参与到高职高专教材的建设中来。在组织教材过程中采取了“请进来,走出去”的工作方法。即:由教育界的专家、领导和一线的教师,以及企事业从事人力资源工作的人员组成顾问班子,充分分析我国各地区的经济发展、产业结构以及人才需求现状,研究培养国家紧缺人才的关键要素,寻求切实可行的教学方法、手段和途径。

通过研讨认识到,我国幅员辽阔,各地区的产业结构有明显的差异,经济发展也不平衡,各地区对人才的实际需求也有所不同。相应地,相同专业和相近专业,不同地区的教学单位在培养目标和培养内容上也各有自己的定位。鉴此,适应教育现状的教材建设应该具有多层次的设计。

为了使教材的编写能针对受教育者的培养目标,出版社的编辑分不同地区逐所学校拜访校长、系主任和老师,深入到高职高专学校及相关企事业,广泛、深入地教学第

一线的老师、用人单位交流,掌握了不同地区、不同类型的高职高专院校的教师、学生和教学设施情况,清楚了各学校所设专业的培养目标和办学特点,明确了用人单位的需求条件。各区域编辑对采集的数据进行统计分析,在相互交流的基础上找出各地区、各学校之间的共性和个性,有的放矢地制定选题项目,并进一步向老师、教育管理者征询意见,在获得明确指导性意见后完成“高职高专规划教材”策划及教材的组织工作:

- 第一批“高职高专规划教材”包括三个学科大系:经济管理、信息技术、建筑。
- 第一批“高职高专规划教材”在注意学科建设完整性的同时,十分关注具有区域人才培养特色的教材出版。
- 第一批“高职高专规划教材”组织过程中,正值高职高专学制从3年制向2年制转轨,教材编写将其作为考虑因素,要求提示不同学制的讲授内容。
- 第一批“高职高专规划教材”编写
 - ◆ 强调以就业岗位对知识和技能需求下的教材体系的系统性、科学性和实用性。
 - ◆ 强调教材以实例为先,应用为目的;围绕应用讲理论,取舍适度,不追求理论的完整性。
 - ◆ 强调提出问题→解决问题→归纳问题的教、学法,培养学生触类旁通的实际工作能力。
 - ◆ 强调课后作业和练习(或实训)真正具有培养学生实践能力的作用。

在“高职高专规划教材”编委的总体指导下,第一批各科教材基本是由系主任,或从教学一线中遴选的骨干教师执笔撰写。在每本书主编的严格审读及监控下,在各位老师的辛勤编撰下,这套凝聚了所有作者及参与研讨的老师们的经验、智慧和资源,涉及三个大的学科近200种的高职高专教材即将面世。我们希望经过近一年的努力,我们奉献给读者的是他们渴望已久的适用教材。同时,我们也清醒地认识到,“高职高专”是正在探索中的教育,加之我们的水平和经验有限,教材的选题和编辑出版会存在许多不尽人意的地方,真诚地希望得到老师和学生的批评建议,以利今后改进,为繁荣我国的高职高专教育不懈努力。

科学出版社

2004年6月1日

前 言

在计算机及其应用的各个领域,都会用到各种各样的数据结构,学会分析研究计算机加工对象的特性,选择合适的数据结构和存储表示,以及编写相应的解题方法(即算法),是计算机专业人才所必不可少的知识。因此“数据结构”是计算机专业教学计划中的核心课程之一。

本书是针对高职高专计算机应用及相关专业编写的,本书基本上覆盖了数据结构的主要内容。全书共分10章。第1章绪论,主要介绍数据结构的基本概念、算法的分析。第2至第7章分别讨论了线性表、栈和队列、串、数组和广义表、树以及图基本类型的数据结构,对每一种数据结构,都讨论了逻辑结构、存储结构以及在各种存储结构下相应的算法。第8章和第9章讨论了排序和查找,并介绍了几种常用的查找和排序的算法。第10章主要介绍实验和上机指导,对本书的每一个内容进行综合练习。本书中的算法都是采用C语言来描述的。

本书可作为高职高专在校生的教材,对三年制学校讲授为64~74学时,对二年制学校讲授为54~64学时,图论中的拓扑排序可以作为选修内容。本书简单易懂,并配有大量的习题,所以,也可以作为计算机专业自修教材。

本书由方风波、王巧莲主编。其中,第1、2、3章由方风波编写;第4、5、8章由黄鹤鸣编写;第6、7、9、10章由王巧莲编写;李军、李太芳、耿杰参加了部分章节的编写和修改,协助主编做了一些前期和后期工作,最后由王巧莲统编全稿,本书由杜友福教授主审。

由于作者水平有限,书中难免出现错误和失误,希望读者批评指正。

作 者

2004年6月

目 录

第1章 绪论	1
1.1 什么是数据结构	1
1.1.1 数据结构的定义	2
1.1.2 数据结构类型	3
1.1.3 数据结构和数据类型	4
1.2 算法及其描述	5
1.2.1 什么是算法	5
1.2.2 算法描述	6
1.3 算法分析	6
1.3.1 时间复杂度	6
1.3.2 空间复杂度	8
1.3.3 算法分析实例	8
1.4 本章小结	9
习题	9
第2章 线性表	12
2.1 线性表及其逻辑结构	12
2.1.1 线性表的定义	12
2.1.2 线性表的操作	13
2.2 线性表的顺序存储结构	14
2.2.1 线性表的顺序存储——顺序表	14
2.2.2 顺序表基本操作的实现	15
2.2.3 顺序表的应用举例	17
2.3 线性表的链式存储	18
2.3.1 线性表的链式存储——链表	18
2.3.2 单链表基本操作的实现	19
2.4 单向循环链表	24
2.5 双向循环链表	25
2.5.1 双向链表	25
2.5.2 双向循环链表	25
2.6 一元多项式的存储和运算	26
2.7 单链表应用举例	29
2.8 本章小结	30
习题	31

第 3 章 栈和队列	33
3.1 栈	33
3.1.1 栈的定义和基本运算	33
3.1.2 栈的顺序存储及其基本操作的实现	34
3.1.3 栈的链式存储及其基本操作的实现	37
3.1.4 栈的应用举例	38
3.2 队列	41
3.2.1 队列的定义及运算	41
3.2.2 队列的顺序存储及其基本操作的实现	42
3.2.3 队列的链式存储及其基本操作的实现	45
3.3 本章小结	46
习题	47
第 4 章 串	49
4.1 串及其操作	49
4.1.1 串的逻辑结构	49
4.1.2 串的基本运算	50
4.2 串的存储结构	50
4.2.1 顺序存储结构及其运算	51
4.2.2 链式存储结构及基本运算的实现	52
4.3 串的模式匹配运算	56
4.3.1 BF (Brute-Force) 算法	56
4.3.2 无回溯的模式匹配 (KMP) 算法	57
4.4 本章小结	58
习题	59
第 5 章 数组和广义表	61
5.1 数组	61
5.1.1 数组的定义	61
5.1.2 数组的顺序存储结构	62
5.1.3 数组的基本操作的实现	63
5.2 稀疏矩阵	64
5.2.1 稀疏矩阵的定义	64
5.2.2 稀疏矩阵的顺序存储结构及基本算法	64
5.2.3 稀疏矩阵的链式存储结构及基本算法	66
5.3 广义表	69
5.3.1 广义表的定义	69
5.3.2 广义表存储结构	70
5.3.3 广义表的基本操作	71
5.4 本章小结	73
习题	73

第6章 树	75
6.1 树的定义和基本操作	75
6.1.1 树的定义	75
6.1.2 树的基本术语	76
6.1.3 树的基本操作	77
6.2 二叉树	77
6.2.1 二叉树的定义及其基本操作	77
6.2.2 二叉树的重要性质	78
6.2.3 二叉树的存储结构	80
6.3 遍历二叉树	83
6.3.1 二叉树遍历的递归算法	84
6.3.2 二叉树遍历的非递归算法	86
6.4 树和森林	87
6.4.1 树的存储结构	87
6.4.2 树与二叉树的转换	89
6.4.3 森林与二叉树的转换图	90
6.5 树的应用	91
6.5.1 二叉排序树	91
6.5.2 哈夫曼树	96
6.6 本章小结	98
习题	99
第7章 图	102
7.1 图的基本概念	102
7.1.1 图的定义	103
7.1.2 图的基本术语	103
7.2 图的存储结构	105
7.2.1 邻接矩阵	105
7.2.2 邻接表	106
7.3 图的遍历	107
7.3.1 深度优先遍历	108
7.3.2 广度优先遍历	110
7.4 生成树	112
7.4.1 概念	112
7.4.2 最小生成树	112
7.5 最短路径	117
7.5.1 求某源点到其余各顶点的最短路径	117
7.5.2 每对顶点之间的最短路径	119
7.6 拓扑排序	120
7.6.1 顶点活动网 (AOV 网)	120

7.6.2 拓扑排序·····	121
7.7 本章小结·····	123
习题·····	124
第8章 排序 ·····	127
8.1 插入排序·····	127
8.1.1 直接插入排序·····	127
8.1.2 希尔排序·····	128
8.2 交换排序·····	129
8.2.1 冒泡排序·····	129
8.2.2 快速排序·····	130
8.3 选择排序·····	132
8.4 归并排序·····	133
8.5 本章小结·····	135
习题·····	136
第9章 查找 ·····	138
9.1 线性表查找·····	138
9.1.1 顺序查找·····	138
9.1.2 折半查找·····	139
9.1.3 分块查找·····	140
9.2 哈希表查找·····	141
9.2.1 哈希表定义·····	141
9.2.2 哈希函数的构造·····	141
9.2.3 哈希冲突解决办法·····	142
9.3 本章小结·····	143
习题·····	144
第10章 实验内容与上机指导 ·····	146
10.1 顺序表及其运算·····	146
10.2 链表及其运算·····	147
10.3 栈的运算·····	151
10.4 队列的运算·····	153
10.5 串的运算·····	156
10.6 二叉树的应用·····	157
10.7 图的存储与遍历·····	161
10.8 排序·····	165
10.9 查找·····	167
主要参考文献 ·····	172

第 1 章 绪 论



知识点

- 数据结构的逻辑结构、存储结构和操作等的相互关系
- 数据结构的类型
- 数据结构和数据类型的区别及联系
- 算法的定义及特征
- 算法的时间复杂度分析



难点

- 数据结构的定义
- 算法的定义
- 算法的时间复杂度分析



要求

掌握:

- 数据的逻辑结构、存储结构和操作的相互关系
- 数据结构的类型(即线性结构、树形结构和图形结构)的区别
- 算法时间复杂度分析

了解:

- 数据结构和数据类型的区别及联系
- 数据结构的这门课的重要性

1.1 什么是数据结构

“数据结构”是计算机专业的基础课之一,是一门十分重要的核心课程。计算机的所有系统软件和应用软件都要用到各种类型的数据结构。要想编出“好”的程序,仅仅学习计算机语言是不够的,必须扎实地掌握数据结构的基本知识和基本技能。要想学好计算机专业的其他课程,如操作系统、数据库原理与应用、软件工程等,也应具备数据结构的基础知识。

计算机的迅猛发展早已使其摆脱了单纯的科学计算,而计算机的普及和发展,使人们要求计算机能处理的问题愈来愈多,愈来愈复杂,问题的规模愈来愈大。要使计算机发挥更大的作用,人们必须设计出更好的程序来。要做到这一点,就应该掌握一系列程序设计知识,其中数据结构的知识是必不可少的。因为人们着手处理问题时,必须分析问题涉及到哪些数据,这些数据具有什么性质,数据之间有什么关系,采用什么方式存储数据并体

现出它们之间的关系,所涉及的问题又包含哪些运算,可采用什么算法。这些正是数据结构这门学科所研究的内容。

1.1.1 数据结构的定义

数据(data)是人们利用文字符号、数字符号以及其他规定的符号对现实世界的事物及其活动所做出的抽象描述。它是计算机程序处理的对象。在计算机科学中,数据的含义极为广泛,如图像、声音、哲学概念、各种感学都可以通过编码而归于数据的范畴。例如,文字、数字和符号都是数据;面包质量管理(针对软硬程序),可以设编码:软=1,硬=2,比较软=1.2 这些对计算机来说都是数据。

数据元素(data element)是数据的基本单位,相当于“记录”。在计算机程序中通常作为一个整体进行考虑和处理。一个数据元素由若干个数据项组成。数据项是数据不可分割的最小单位,也是数据集合的最小可命名单位,相当于记录中的“域”。例如:在学生档案管理系统中,可以把学生的有关信息作为一个数据元素,它由学号、姓名、年龄等数据项组成。

数据对象(data object)是性质相同的数据元素的集合,是数据的一个子集。例如,整数数据对象是集合 $N = \{0, \pm 1, \pm 2, \dots\}$, 字母字符数据对象是集合 $C = \{'A', 'B', \dots, 'Z'\}$ 。

数据结构(data structure)是指数据以及相互之间的关系,可以看作是相互之间存在一种或多种特定关系的数据元素的集合。这些特定关系包括:

- 数据元素之间的逻辑关系,即数据的逻辑结构。
- 数据元素及其关系在计算机存储器中的存储方式,即数据的存储结构,也就是数据元素的物理结构。
- 施加在数据上的操作,即数据的运算。

数据的逻辑结构是从逻辑关系上描述数据,它与数据的存储无关,是独立于计算机的。因此,数据的逻辑结构可以看作是从具体问题抽象出来的数学模型。数据的存储结构是逻辑结构用计算机语言的实现(亦称为映象),它是依赖于计算机语言的。对机器语言而言,存储结构是具体的,但本书只在高级语言的层次上来讨论存储结构。数据的运算是定义在数据的逻辑结构上的,每种逻辑结构都有一个运算的集合。例如,最常用的运算有:查找、插入、删除、更新、排序等。这些运算实际上是在抽象的数据上所施加的一系列抽象的操作。所谓抽象的操作,是指只知道这些操作是“做什么”,而无须考虑“如何做”。只有确定了存储结构之后,才考虑如何具体实现这些运算。

【例 1.1】 有一个学生表,如表 1.1 所示。这个表中的数据元素是学生记录,每个数据元素由四个数据项(即学号、姓名、性别和班号)组成。

我们把表 1.1 称为一个数据结构。表中的每一行是一个数据元素(或记录),它由学号、姓名、性别、班号四个数据项组成。该表中数据元素之间的逻辑关系是:对表中任一结点,与它相邻且在它前面的结点(亦称为直接前趋)最多只有一个;与表中任一结点相邻且在其后的结点(亦称为直接后继),也最多只能有一个。表中只有第一个结点没有直接前趋,故称为开始结点;只有最后一个结点没有直接后继,故称为终端结点。例如,表中“刘红”所在结点的直接前趋和直接后继分别是“张斌”和“李秀平”所在的结点,上述结点

间的关系构成了这张学生档案表的逻辑结构。

表 1.1 学生表

学 号	姓 名	性 别	班 号
1	张斌	男	9901
5	刘红	女	9901
6	李秀平	女	9902
7	董志强	男	9902
8	王平银	男	9901
2	张力	男	9901

表的存储结构是指用计算机语言如何表示结点之间的这种关系,即表中的结点是顺序邻接地存储在一片连续的单元之中,还是用指针将这些结点链接在一起?在这张表中,可能要经常查看某一个学生的成绩,当学生退学时要删除相应的结点,进来新学生时要增加结点。究竟怎样进行查找、删除、插入,这就是数据的运算问题。搞清楚了上述三个问题,也就弄清楚了学生档案表这个数据结构。

综上所述,可以将数据结构定义为:按某种逻辑关系组织起来的一批数据,应用计算机语言,按一定的存储结构表示方式把它们存储在计算机的存储器中,并在这些数据上定义了一个运算的集合。

1.1.2 数据结构类型

在不会产生混淆的前提下,常常将数据的逻辑结构简称为数据结构。数据的逻辑结构有四种:

(1)集合。结构中的数据元素之间除了存在一个“属于同一集合”的关系外,没有其他关系存在。

(2)线性结构。结构中的数据元素之间存在一个一对一的关系,如表 1.1 所示。

(3)树形结构。结构中的数据元素之间存在一个一对多的关系。

(4)图(网形结构)。结构中的数据元素之间存在多个多对多的关系。

从集合论的观点出发,可以对数据结构做出形式定义:数据结构是一个二元组:

$$\text{Structure} = (D, R)$$

其中,D 是数据元素的有限集合(数据对象),R 是 D 上关系的有限集。

例如,一个班级:一个辅导员,若干个学生。班级是一个数据结构; $D = \{i, a_1, a_2, \dots, a_n\}$ 是数据元素的集合,其中 i 是辅导员, $a_1, a_2, \dots, a_n$ 为 n 个学生。 $R = \{\langle i, a_1 \rangle, \langle i, a_2 \rangle, \dots, \langle i, a_n \rangle\}$ 是辅导员与学生之间的关系。这是一个一对多的关系,即一个辅导员与多个学生相关。

关系 R 描述的是数据的逻辑结构。为了在计算机中处理这种类型数据,就需要将逻辑结构存储(映像)在存储器中,即存储结构,在这里不仅仅是将数据元素存放在存储器中,而且这种存放数据元素的方法要能表现关系 R。在存储结构中,数据元素之间的关系

的表示方法(即存储结构)有以下四种:

(1)顺序存储。采用连续的一段存储器空间存储数据元素,并借助元素在存储器中的相对位置来表示数据元素之间的逻辑关系。

(2)非顺序存储。将数据元素所占的存储单元分成两部分,一部分存其值,另一部分存放表示数据元素之间关系的指针。

上述称为顺序结构和链式结构。

(3)索引存储。顺序存储和链式存储的组合。

(4)哈希存储。由元素的关键值直接计算出该结点的存储地址。

1.1.3 数据结构和数据类型

数据类型是与数据结构密切相关的一个概念,容易引起混淆。

数据类型是最早出现在高级语言中,是对数据的取值范围、每一数据的结构以及允许施加操作的一种描述,它刻画了程序中操作对象的特性。每一种程序都定义有自己的数据类型,在用程序语言编写的程序中,每个变量、常量或表达式都有一个它所属的确定的数据类型。

数据类型可分为简单类型和结构类型两种。简单类型中的每个数据(即简单数据)都是无法再分割的整体,如一个整数、实数、字符、指针、枚举量等都是无法再分割的整体,所以它所属的类型均为简单类型。结构类型由简单类型按照一定的规则构造而成,并且结构类型中可以包含结构类型。所以一种结构类型中的数据可以分解为若干个简单或结构类型,每个结构数据仍可再分。例如 C 语言中的数组是一种结构类型。

数据类型也可以被定义为一种数据结构和能够对该数据结构进行操作的集合。对于简单类型,其数据结构就是相应取值范围内的所有数据,每一个数据是不可分割的独立整体,因而数据值内部就无结构可言。对于结构类型,其数据结构就是相应的元素和元素之间所含关系的集合。

总之,数据结构是指计算机处理的数据元素的组织形式和相互关系,而数据类型是某种程序设计语言中已实现的数据结构。在程序设计语言类型支持下,就可以根据从问题中抽象出来的各种类型模型,逐步构造出描述这些数据模型的各种新的数据结构。

下面总结 C 语言常用的数据类型。

1. C 语言的基本类型

C 语言中的基本数据类型有 int 型、float 型和 char 型。int 型可以有三种形式:short, long 和 unsigned。float 型有三种形式:float, double 和 long double。

2. C 语言的指针类型

C 语言允许直接对存放变量的地址进行操作。如定义 int i, 则 &i 表示变量 i 的地址,也称作指向变量 i 的指针。存放地址的变量称为指针变量。

3. C 语言的数组类型

数组是同一类型的一组有序数据的集合。数组有一维数组和多维数组。数组名标识

一个数组,下标指示一个数组元素在该数组中的顺序位置。

数组下标的最小值称为下界,在C语言中总是为0。数组下标的最大值称为上界,在C语言中数组上界为数组定义值减1,例如, `int a[10]`,定义了包含10个整数的数组a,其下标的最大值为9。

4. C语言中的结构体类型

结构体由一组称为结构体成员的项组成,每个结构体成员都有自己的标识符。例如:

```
struct teacher
{
    int no ;
    char name[8];
    int age ;
};
```

定义了一个结构体类型 `teacher`,可以利用下句来定义该类型变量 `t1,t2`。

```
struct teacher t1,t2;
```

5. C语言中的共用体类型

共用体是把不同的成员组织为一个整体,它们在存储器中共享一段存储单元,但以不同的方式解释不同的成员。例如:

```
union tag
{
    int no1;
    char no2[2];
};
```

定义了一个共用体类型 `tag`,整型变量 `no1` 和字符型数组变量 `no2` 共享相同的存储单元。

6. C语言中的自定义类型

C语言中允许使用 `typedef` 关键字来定义同名的数据类型名。例如:

```
typedef int integer ;
```

将 `int` 类型与 `integer` 等同起来。

1.2 算法及其描述

1.2.1 什么是算法

算法(algorithm)是对特定问题求解的一种描述,它是指令的有限序列,其中每一条指令表示一个或多个操作。此外,一个算法还具有下列五个重要特性:

(1)有穷性。一个算法必须总是对任何合法的输入值在执行有穷步之后结束,且每一步都可在有穷时间内完成。

(2)确定性。一个算法中每一条指令必须有确定的含义,不会产生二义性。