

● 计算机精品教材

SHUJU JIEGOU

# 数据结构 (C语言版)

洪运国 编著



上海交通大学出版社  
SHANGHAI JIAO TONG UNIVERSITY PRESS

计算机精品教材

# 数据结构 (C语言版)

洪运国 编著



上海交通大学出版社  
SHANGHAI JIAO TONG UNIVERSITY PRESS

## 内容提要

本书以项目方式全面系统地介绍了各种类型的数据结构,并从逻辑结构、存储结构和基本操作几个方面进行了详细阐述。全书共八个项目,分别介绍了线性表、栈、队列、串、数组、树、图等基本类型的数据结构,以及查找、排序技术。

全书语言简练、结构合理、图示丰富,给出了众多使用类C语言描述的算法,从而加深读者对抽象概念的理解。与此同时,每个项目都给出了与各项目内容相关的,应用基本算法的众多上机实训,并给出了注释详尽的源代码和测试结果,供读者上机练习之用。另外,每个项目都给出了精心设计的习题和上机操作题,从而让读者强化理解和巩固本项目所学知识。

本书可作为高等院校计算机、信息技术类专业的专业课教材,也可作为从事软件设计和开发的相关工程技术人员的参考书。

## 图书在版编目(CIP)数据

数据结构 : C语言版 / 洪运国编著. — 上海 : 上海交通大学出版社, 2016

ISBN 978-7-313-15217-6

I. ①数… II. ①洪… III. ①数据结构—高等学校—教材②C语言—程序设计—高等学校—教材 IV. ①TP311.12②TP312

中国版本图书馆CIP数据核字(2016)第156395号

## 数据结构(C语言版)

编 著: 洪运国

出版发行: 上海交通大学出版社

邮政编码: 200030

出 版 人: 韩建民

印 制: 三河市祥达印刷包装有限公司

开 本: 787mm×1092mm 1/16

版 次: 2016年7月第1版第1次印刷

书 号: ISBN 978-7-313-15217-6/TP

定 价: 48.00元

地 址: 上海市番禺路951号

电 话: 021-64071208

经 销: 全国新华书店

印 张: 20 字 数: 388千字

版权所有 侵权必究

告读者: 如发现本书有印装质量问题请与发行部联系

联系电话: 010-62137141



# 前 言

随着人类进入信息社会,计算机技术已经渗透到社会生活的各个领域,整个社会对具有计算机专业技能和实用技术的应用型人才的需求更加迫切。如何培养符合时代要求、适应就业市场需要的优秀人才是全社会尤其是高等院校面临的一项紧迫任务。

数据结构是高等院校计算机和信息技术类专业的一门重要专业基础课,其主要研究如何存储和组织数据以及如何处理数据的问题。读者在初步掌握了计算机的基础知识和一种程序设计语言之后(本书主要使用 C 语言),通过学习本课程可以明显提高编程水平和解决实际问题的能力。因此,对于软件开发人员来说,仅懂得开发工具的语言规则和使用是远远不够的,还应该具备数据抽象能力和算法设计能力,而这些正是数据结构所要求掌握的。

由于数据结构中对存储结构和算法的描述比较抽象,所以这门课程一直是高等院校学生学习的难点。为此,作者根据教学大纲要求,按照学生的认知规律,广泛吸收各家长处,本着求实、创新的精神编写了这本应用型教材《数据结构(C 语言版)》,以满足高等院校学生和社会各界的迫切需要。

## 本书内容

本书以项目方式全面系统地介绍了各种类型的数据结构,并从逻辑结构、存储结构和基本操作几个方面进行了详细阐述。全书共八个项目,分别介绍了线性表、栈、队列、串、数组、树、图等基本类型的数据结构,以及查找、排序技术。

## 本书特色

与目前大部分数据结构教材相比,本书主要有如下几个特点:

(1) 采用项目教学法,以引导案例入手。为了适应目前的教学改革要求,本书按照新型项目教学法的要求,每个项目都以一个引导案例入手,从而增强学生的学习兴趣,让学生带着问题学。

(2) 语言简练、结构合理、图示丰富。本书在理论知识的讲解上力求语言简明扼要、通俗易懂,在结构安排上,由浅入深、循序渐进。与此同时,对于难以理解的内容均配以图示说明,从而使读者更容易理解一些抽象的概念和算法。

(3) 理论与实践密切结合、相得益彰。为培养读者的动手实践能力,书中给出了许多典型的算法,并以类 C 语言对其进行描述。同时,通过对算法思路的分析,还可培养读者的

算法设计能力和创造性思维方法, 并能够举一反三、触类旁通。

(4) 丰富的上机实训。为了便于教学, 每个项目都安排了多个上机实训, 提供了每个项目应用基本算法的完整程序, 并给出了注释详尽的源代码和测试结果, 从而供学生上机练习之用。

(5) 精心设计的习题与上机操作题。每个项目最后都给出了精心设计的习题和上机操作题, 基本覆盖了读者学习本项目应掌握的重点内容, 是对本项目内容的强化和教学成果的检验, 对巩固所学知识很有帮助。

### 本书读者对象

本书可作为高等院校计算机、信息技术类专业的专业课教材, 另外, 由于本书通俗易懂, 便于自学, 故也可供从事计算机软件设计和开发的相关工作人员参考。

### 教学资源下载

本书配有精心制作的教学课件, 供教师上课使用, 从而最大限度地减轻教师的负担。另外, 书中涉及的全部程序源代码、习题参考答案等均可到网站 (<http://www.bjjqe.com>) 下载。如果读者在学习过程中有什么疑问, 也可登录该网站寻求帮助。

本书由洪运国编著, 尽管在编写时已付出巨大努力, 但由于编写时间仓促, 编者水平有限, 书中疏漏与不当之处敬请广大读者批评指正。另外, 本书在写作时参考了大量的文献资料, 在此谨向参考过的中外文献的作者致以诚挚的谢意。

编者  
2016年6月

# 目 录

项目一 数据结构导论	1
任务一 了解数据结构的基本知识	1
1.1 基本术语	2
1.2 数据的逻辑结构	3
1.3 数据的存储结构	3
1.4 数据类型	4
任务二 了解算法的基本知识	6
2.1 算法的概念	6
2.2 算法的特性	6
2.3 算法的描述方法	6
2.4 算法设计的要求	8
2.5 算法性能分析	9
2.6 类 C 语言简介	10
实训 1-1 将百分制成绩转换成五级分制成绩	10
实训 1-2 求一组数中的最大数和最小数	13
项目小结	14
习题与上机操作	14
项目二 线性表	16
任务一 了解线性表的定义和基本操作	16
1.1 线性表的定义	16
1.2 线性表的基本操作	17
任务二 掌握顺序表的结构特点与基本操作	18
2.1 顺序表的结构特点	18
2.2 顺序表的基本操作	19
任务三 掌握链表的结构特点与基本操作	23
3.1 单链表的结构特点	23
3.2 单链表的基本操作	25
3.3 静态链表及其基本操作	31
3.4 循环链表及其基本操作	33
3.5 双向链表及其基本操作	34
实训 2-1 学生成绩统计	37
实训 2-2 求两个集合的交集	41



实训 2-3 约瑟夫环问题	44
实训 2-4 一元多项式运算	46
项目小结	52
习题与上机操作	52
项目三 栈和队列	55
任务一 了解栈的定义、基本操作和存储结构	55
一、栈的定义及其基本操作	55
二、栈的顺序存储结构	56
三、栈的链式存储结构	60
四、栈在递归中的应用	64
任务二 掌握队列的定义、基本操作和存储结构	65
一、队列的定义及其基本操作	65
二、队列的顺序存储结构	66
三、队列的链式存储结构	69
实训 3-1 算术表达式求值	71
实训 3-2 汉诺塔问题	78
实训 3-3 迷宫求解	80
实训 3-4 打印杨辉三角	85
项目小结	89
习题与上机操作	89
项目四 串和数组	92
任务一 了解串的定义、基本操作和存储结构	92
一、串的定义及其基本操作	92
二、定长顺序存储结构	94
三、堆存储结构	100
四、块链存储结构	104
任务二 掌握数组的定义、基本操作和存储结构	105
一、数组的定义及其基本操作	105
二、数组的顺序存储结构	106
任务三 掌握特殊矩阵和稀疏矩阵的压缩存储方法	107
一、特殊矩阵的压缩存储	108
二、稀疏矩阵的压缩存储	110
任务四 了解广义表的定义和存储结构	112
一、广义表的定义	112
二、广义表的存储结构	113
实训 4-1 串的查找与替换	114
实训 4-2 串的模式匹配	116
实训 4-3 稀疏矩阵的转置	121

实训 4-4 稀疏矩阵相加	125
项目小结	132
习题与上机操作	132
<b>项目五 树与二叉树</b>	134
<b>任务一 了解树的定义和基本操作</b>	134
一、树的定义和表示方法	134
二、树的基本术语	135
三、树的基本操作	136
<b>任务二 掌握二叉树的特点和常用操作</b>	137
一、二叉树的定义	137
二、二叉树的基本操作	138
三、二叉树的性质	138
四、二叉树的存储结构	140
五、遍历二叉树的方法	143
六、线索二叉树的定义和基本操作	147
<b>任务三 掌握树和森林的存储结构与相关操作</b>	151
一、树的存储结构	151
二、树、森林和二叉树的转换	154
三、树和森林的遍历	157
<b>任务四 了解哈夫曼树的定义、构造方法及其应用</b>	157
一、哈夫曼树的定义	157
二、哈夫曼树的构造	158
三、哈夫曼树编码	159
实训 5-1 创建一棵二叉树并求它的高度	160
实训 5-2 判断二叉树是否为完全二叉树	163
实训 5-3 构造哈夫曼编码	166
实训 5-4 8 枚硬币的真假判定问题	171
项目小结	175
习题与上机操作	175
<b>项目六 图</b>	178
<b>任务一 了解图的定义、基本术语和基本操作</b>	178
一、图的定义	178
二、图的基本术语	179
三、图的基本操作	181
<b>任务二 掌握图的存储结构</b>	182
一、邻接矩阵表示法	182
二、邻接表表示法	184
三、十字链表表示法	186





四、邻接多重表表示法	188
任务三 掌握图的遍历方法	190
一、深度优先遍历	190
二、广度优先遍历	191
任务四 了解图的相关应用	192
一、生成树和最小生成树	192
二、最短路径	196
三、拓扑排序	199
四、关键路径	203
实训 6-1 图的建立与遍历	207
实训 6-2 拓扑排序	211
实训 6-3 公共汽车的换乘问题	216
项目小结	220
习题与上机操作	220
项目七 查找	223
任务一 了解查找的相关术语	223
任务二 掌握静态查找方法	224
一、顺序查找	225
二、折半查找	226
三、索引顺序查找	230
任务三 掌握动态查找方法	232
一、二叉排序树	233
二、平衡二叉树	241
任务四 掌握哈希查找方法	252
一、哈希表的基本概念	252
二、构造哈希函数的方法	253
三、处理冲突的方法	256
四、哈希表的查找及分析	259
实训 7-1 利用折半查找法查找学生成绩	261
实训 7-2 创建二叉排序树并查找	264
实训 7-3 利用哈希表查找员工信息	267
项目小结	270
习题与上机操作	270
项目八 排序	272
任务一 了解排序的相关概念	272
任务二 掌握插入排序方法	274
一、直接插入排序	274
二、折半插入排序	276

三、希尔排序	277
任务三 掌握交换排序方法	279
一、冒泡排序	279
二、快速排序	281
任务四 掌握选择排序方法	284
一、直接选择排序	284
二、树形选择排序	285
三、堆排序	287
任务五 掌握归并排序和基数排序方法	292
一、归并排序	292
二、基数排序	294
实训 8-1 利用直接插入排序方法排序学生成绩表	299
实训 8-2 利用冒泡排序方法排序学生成绩表	302
实训 8-3 利用直接选择排序方法排序学生成绩表	304
实训 8-4 利用堆排序方法排序学生成绩表	305
项目小结	308
习题与上机操作	308
参考文献	310

## 任务一 了解数据结构的基本知识

随着计算机应用的日益深入,人们开始借助计算机来解决很多现实问题。为此,我们需要解决如下几个问题:

(1) 如何利用计算机能理解的数据形式来描述现实问题。例如,要设计一个学生成绩管理信息系统,首要的问题就是要设计一个合理的数据结构来存储每个学生的学号、姓名、成绩等信息。

(2) 针对所要解决的问题确定算法和编制程序。例如,就学生成绩管理信息系统而言,我们首先要考虑数据的输入、删除、修改、插入、排序、查询等管理问题。那么,如何设计一个排序算法,使排序、查询等操作的效率最高,就是数据结构要研究的问题了。

在计算机专业中,数据结构是一门综合性的专业基础课,是介于数学、计算机和计算机应用的一个交叉学科,它主要研究非数值计算的程序设计问题中计算机操作对象,以及这些对象之间的相互关系等。

# 项目一 数据结构导论

## 【引例】

根据表 1-1, 编制一个程序将学生百分制成绩转换成相应的五级分制成绩。总体而言, 这个程序非常简单。但是采用的算法不同, 程序的运行效率会有很大的差别, 而算法正是数据结构这门课程要研究的主要内容之一。

表 1-1 学生成绩转换方式

分数	0~59	60~69	70~79	80~89	90~100
比例数	0.05	0.15	0.40	0.30	0.10
等级	bad (不及格)	pass (及格)	general (中等)	good (良好)	excellent (优秀)

## 【学习目标】

- ◆ 了解数据结构课程的研究内容和学习目的。
- ◆ 理解数据的逻辑结构、存储结构和数据类型等概念。
- ◆ 了解算法的概念、特性、描述方法和设计要求。
- ◆ 了解对算法性能评价的重要指标。

## 任务一 了解数据结构的基本知识

随着计算机应用的日益深入, 人们开始借助计算机来解决很多现实问题。为此, 我们需要解决如下几个问题:

(1) 如何利用计算机能理解的数据形式来描述现实问题。例如, 要设计一个学生成绩管理信息系统, 首要的问题就是要设计一个合理的数据结构来存储每个学生的学号、姓名、班级、成绩等信息。

(2) 如何根据要解决的问题确定算法和编制程序。例如, 就学生成绩管理信息系统而言, 我们需要解决数据的输入、删除、修改、插入、排序、查询等管理问题。那么, 如何通过设计合理的算法使排序、查询等操作的效率最高, 就是数据结构要研究的问题了。

在计算机学科中, 数据结构是一门综合性的专业基础课, 是介于数学、计算机和计算机软件之间的一门核心课程, 它主要研究非数值计算的程序设计问题中计算机的操作对象, 以及这些对象之间的关系和运算等。



## 一、基本术语

为了便于大家学习后面的内容，下面首先介绍几个基本术语。

### 1. 数据

**数据 (data)** 是对客观事物的符号表示，在计算机科学中，是指所有能输入到计算机中并被计算机程序处理的符号的总称。它是计算机程序加工的“原料”。例如，一个利用数值分析方法解代数方程的程序，其处理的对象是整数和实数；一个编译程序或文字处理程序，其处理对象是字符串。因此，对计算机科学而言，数据的含义极为广泛，如图像、声音等都可以通过编码而归之于数据的范畴。

### 2. 数据元素

**数据元素 (data element)** 是数据的基本单位，一个数据元素可由若干个数据项组成，此时的数据元素通常称为记录 (record)。如表 1-2 学生基本信息表中的数据，每一个学生的信息记录就是一个数据元素。

表 1-2 学生基本信息表

学号	姓名	性别	出生日期	住址	专业	成绩
200903105	张强	男	90/06	山东泰安	电子信息技术	550
20091011	李红	女	90/11	河南信阳	建筑	480
20093104	李华	男	91/01	福建福州	机械制造	440
.....	.....	.....	.....	.....	.....	.....

### 3. 数据项

**数据项 (data item)** 是组成数据元素不可再分的最小单位，如学生信息记录中的学号、姓名等。

### 4. 数据对象

**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，{2, 4, 6, 8, 10} 是整数的一个数据对象。在这个数据对象中，每一个数据元素都是性质相同的整数，是有限个整数数据元素的集合，是整数的一个子集。

### 5. 数据结构

**数据结构 (data structure)** 是相互之间存在一种或多种特定关系的数据元素的集合。结构就是指数据元素之间的相互关系，即数据的组织形式。数据结构一般包括三方面的内容：

① 数据元素之间的逻辑关系，又称数据的逻辑结构。它与数据的存储无关，是独立于计算机的。

② 数据元素及其关系在计算机内存中的表示，又称数据的物理结构，或数据的存储结构。

③ 数据的运算及实现，即可以对数据元素施加的操作，以及这些操作在相应存储结构上的实现。

## 二、数据的逻辑结构

数据的逻辑结构是指数据元素之间逻辑关系的描述。根据数据元素之间关系的不同特性，通常有下列4类基本结构，如图1-1所示。

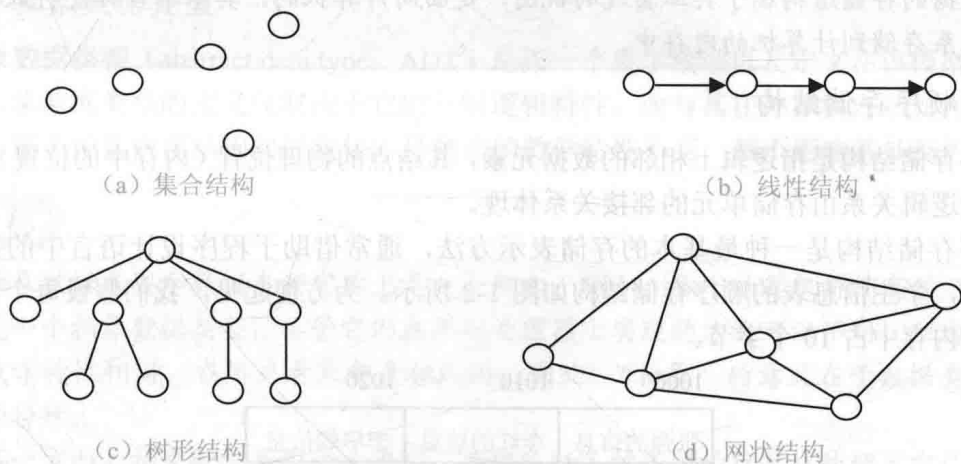


图 1-1 数据的逻辑结构

- **集合**：数据元素之间除了同属于一个集合外，无其他关系。
- **线性结构**：数据元素之间是一一对一的关系。此时每个数据元素都有唯一的前驱（第一个元素除外）和唯一的后继（最后一个元素除外）。例如，列车中各车厢之间的关系、排队买票人之间的关系、一碟盘子中各盘子之间的关系等。
- **树形结构**：数据元素之间是一对多的关系。例如，学校的组织结构、家族关系等。在树形结构中，最上面的结点称为根节点，最下面的结点称为叶子结点。每个结点最多只能有一个父结点，但可以有多个子结点。
- **图状结构或网状结构**：数据元素之间是多对多的关系。例如，城市公共交通网、计算机网络等。此时每个结点都可以有多个前驱或后继。

## 三、数据的存储结构

数据的存储结构或物理结构是指数据在计算机中的表示或存储方式，是逻辑结构在计算机中的实现，包括数据元素的表示和关系的表示。

依据数据元素之间的关系在计算机中的不同表示，有两种不同的存储结构，分别是顺序存储结构和链式存储结构。



数据的逻辑结构是指数据元素之间整体的逻辑关系。例如，一个工厂里的工人之间可能存在着上下级关系、血缘关系、同乡关系、同学关系等。从数据结构的角度来看，所有这些关系都可以抽象为数据元素之间的逻辑关系。数据的逻辑结构属于用户视图，



是面向问题的，反映了数据内部的构成方式。

数据的存储结构是数据及其逻辑结构在计算机中的表示。换言之，存储结构除了存储数据元素之外，必须能隐式或显式地表示数据元素之间的逻辑关系。这样，在逻辑上相邻的数据元素，在存储结构中未必相邻。例如，父子关系可以看成是逻辑关系，在逻辑上相邻，但他们未必生活在同一个地方，在物理上可能生活在不同的城市，甚至不同的国家。

数据的存储结构属于具体实现的视图，是面向计算机的，其基本目标是将数据及其逻辑关系存储到计算机的内存中。

### 1. 顺序存储结构

顺序存储结构是指逻辑上相邻的数据元素，其结点的物理位置（内存中的位置）也相邻，结点间的逻辑关系由存储单元的邻接关系体现。

顺序存储结构是一种最基本的存储表示方法，通常借助于程序设计语言中的数组来实现。例如，学生信息表的顺序存储结构如图 1-2 所示。为方便起见，我们假设每个学生的基本信息在内存中占 10 个字节。

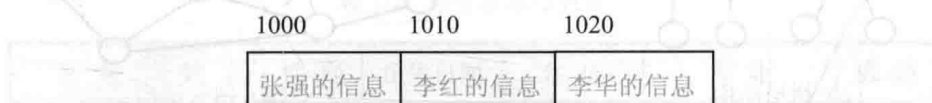


图 1-2 顺序存储结构示例

### 2. 链式存储结构

在链式存储结构中，结点间的逻辑关系是由附加的指针字段来指示的。因此，链式存储结构通常借助于程序设计语言中的指针数据类型来实现。例如，学生信息表的链式存储结构如图 1-3 所示。由此可以看出，在这种存储结构中，逻辑上相邻的数据元素，其物理位置不一定相邻。

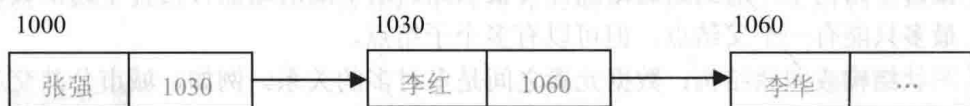


图 1-3 链式存储结构示例

## 四、数据类型

### 1. 数据类型

数据类型 (data type) 是与数据结构密切相关的一个概念，它最早出现在高级程序语言中是用来描述操作对象的特性的。在用高级语言编写的程序中，每个变量、常量或表达式都有一个确定的数据类型。

类型显式或隐含规定了在程序执行期间变量或表达式所有可能的取值范围，以及在这些值上允许进行的操作。因此，数据类型实际上是一个值的集合以及定义在这个值集上的一组操作的总称。例如，C 语言中的整型类型 (int) 可能的取值范围是 -32768~+32767，可执



行的运算有加、减、乘、除、取模等。

按值的不同特性,高级程序语言中的数据类型分为两类:一类是非结构的原子类型,其值是不可分解的,如C语言中的基本类型(整型、实型、字符型和枚举类型)、指针类型和空类型;另一类是结构类型,其值是由若干成分按某种结构组合而成的,因此它是可以分解的,并且它的成分可以是非结构的,也可以是结构的。例如,数组、结构体类型的值由若干分量组成,每个分量可以是整数,也可以是数组等。

## 2. 抽象数据类型

**抽象数据类型**(abstract data type, ADT)是指一个数学模型以及定义在该模型上的一组操作。抽象数据类型的定义仅取决于它的一组逻辑特性,而与其在计算机内部如何表示和实现无关。即不论其内部结构如何变化,只要它的数学特性不变,都不影响其外部的使用。



### 提示

抽象数据类型和数据类型实质上是一个概念。例如,各个计算机都拥有的“整数”类型是一个抽象数据类型,尽管它们在不同处理器上实现的方法可以不同,但由于其定义的数学特性相同,在用户看来都是相同的。因此,“抽象”的意义在于数据类型的数学抽象特性。

另一方面,抽象数据类型含义更广,不仅包括各种不同的计算机处理器中已定义并实现的数据类型(这类数据类型又称固有数据类型),还包括设计软件系统时用户自己定义的复杂数据类型。并且,所定义的数据类型的抽象层次越高,含有该数据类型的软件复用程度就越高。

抽象数据类型是近年来计算机科学中提出的最重要的概念之一,它集中体现了程序设计中一个最基本的原则:通过封装和信息隐蔽,使对象操作的具体实现方法和外部引用相分离。

我们可以用三元组(D, S, P)来描述抽象数据类型。其中,D是数据对象,S是D上的关系集,P是对D的操作集。其格式定义如下:

ADT 抽象数据类型名

{

    数据对象: <数据对象的定义>

    数据关系: <数据关系的定义>

    基本操作: <基本操作的定义>

}ADT 抽象数据类型名

例如,线性表的抽象数据类型描述如下:

ADT Liner\_List

{

    数据元素:  $D = \{a_i \mid a_i \in \text{elemset}, i = 1..n (n \geq 0)\}$

    数据关系:  $S = \{\langle a_{i-1}, a_i \rangle \mid a_{i-1}, a_i \in D, i = 2..n (n \geq 0)\}$

    基本操作:

    InitList (L): 初始化为一个空的线性表。



ListLength (L): 求线性表 L 中元素个数。

GetElem (L, i): 取线性表 L 中的第 i 个元素。

Locate (L, e): 确定元素 e 在线性表 L 中的位置。

InsertList (L, i, e): 在线性表 L 的第 i 个位置之前插入数据元素 e。

DeleteList (L, i): 删除线性表 L 中第 i 个元素。

}ADT Liner\_List

## 任务二 了解算法的基本知识

### 一、算法的概念

在实际生活中,我们做任何事情都是有一定步骤的。例如,要在商场购买某种商品,应首先挑选商品,然后开票、到收银台付款、提货等。因此,简单地说,算法就是指为解决某一问题所采取的方法和步骤。

在计算机领域中,算法是指:根据所要处理的问题,在数据的逻辑结构和物理结构基础上,利用有限步骤内解决这一问题所采用的一组指令序列。

### 二、算法的特性

一个算法应该具有以下特性:

- 有穷性: 一个算法必须在有限步骤之内正常结束,不能形成无穷循环。
- 确定性: 算法中的每一个步骤必须有确定含义,不能有二义性。
- 可行性: 算法中的每一个步骤都应当可以有效执行,并得到确切结果。
- 输入: 有 0 个或多个输入。
- 输出: 至少有一个或多个输出。

在以上 5 个特性中,最基本的特性是有穷性、确定性和可行性。

### 三、算法的描述方法

算法可以使用各种不同的方法来描述,如自然语言、流程图等,下面分别进行介绍。

#### 1. 自然语言

用自然语言描述算法的优点是简单直观且便于人们阅读;缺点是不够严谨,与计算机的具体高级程序设计语言相差很大,需要用户进行转换。例如,要求一组数据中的最大数和最小数。用自然语言描述的算法如下:

- ① 将第 1 个数和第 2 个数相比较,记录最大数与最小数。
- ② 将最大数和最小数与第 3 个数比较,记录最大数与最小数。
- ③ 重复步骤②,依次与剩下的数进行比较,直至结束。

#### 2. 流程图

描述算法还可以使用流程图方式,其特点是直观、简洁、明了。例如,要求一组数据 (n 个数) 中的最大数和最小数,可用图 1-4 所示的流程图描述其算法。



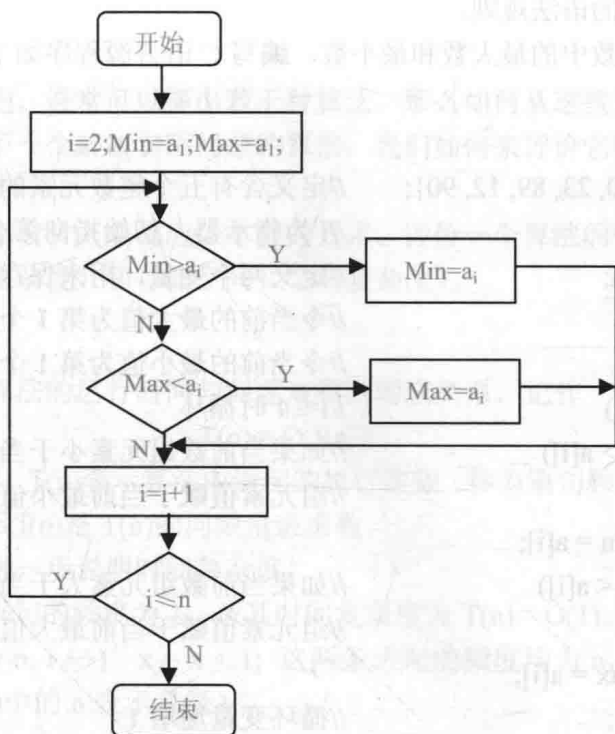


图 1-4 程序流程图

### 3. 伪代码

用传统的流程图和 N-S 图描述算法的优点是直观易懂，但画起来比较费事。另外，由于在设计一个算法时可能需要反复修改，而修改流程图是比较麻烦的。因此，在设计算法时为了方便起见，常用一种称为伪代码的工具。

伪代码是介于自然语言和计算机语言之间的语言，常用文字和符号来表示。它不用图形符号，因此书写方便、格式紧凑，也易于理解，便于向计算机语言算法（即程序）过渡。例如，要求一组数（ $n$  个）中最大数和最小数，用伪代码表示其算法如下。

开始

置  $i$  的初值为 2

置  $Min$  和  $Max$  的初值为  $a_1$

当  $i \leq n$  时，执行如下操作

    如果  $Min > a_i$ ，则使  $Min = a_i$ ；

    如果  $Max < a_i$ ，则使  $Max = a_i$ ；

$i = i + 1$

（循环体到此结束）

打印  $Max$  和  $Min$  值

结束

### 4. 程序设计语言

用计算机编程语言表示算法实际上就是用某种语言来编制程序。和伪代码不同，编程必