21世纪高职高专计算机系列教材

数据结构

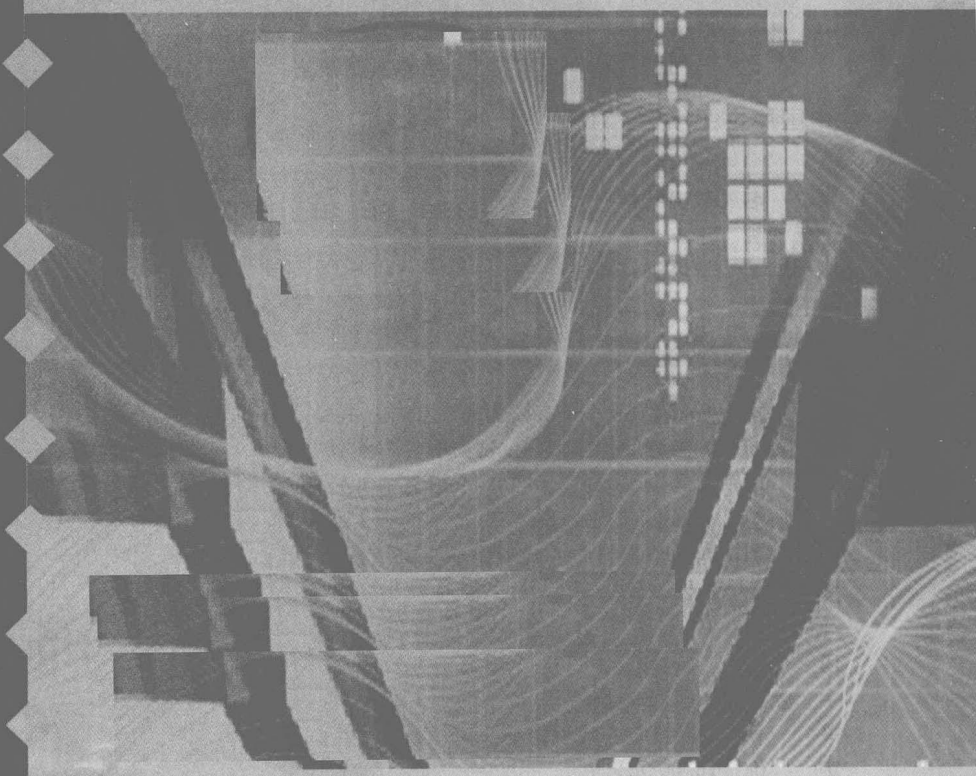
主　编　王忠义
主　审　龚尚福

西安交通大学出版社

21世纪高职高专计算机系列教材

数据结构

主　编　王忠义
编　者　张小艳　王　敏
　　周筱媛　刘文哲
主　审　龚尚福



西安交通大学出版社
·西　安·

内 容 提 要

本书介绍了数据结构的一般概念和算法分析的初步知识，介绍了线性表、栈、队列、串、数组、树、图等数据结构的基本概念、存储结构和不同存储结构上的一些操作的实现算法，讨论了查找和排序的各种方法，简单分析了各种算法的时间性能。本书采用C语言描述算法。

本书是为高职高专计算机及相关专业学生编写的教材。根据高职高专注重实践的特点，书中尽量避免抽象理论的介绍，通过实例使学生理解数据结构的基本概念。书中安排多个综合实例和完整的C语言程序供学生上机实习参考，每章带有小结和适量的习题。

图书在版编目(CIP)数据

数据结构 / 王忠义主编 . —西安：西安交通大学出版社，2003.12

(21世纪高职高专计算机系列教材)

ISBN 7-5605-1783-8

Ⅰ.数... Ⅱ.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(2003)第108802号

书　　名	数据结构
主　　编	王忠义
策划编辑	贺峰涛　屈晓燕
文字编辑	邹　林
出版发行	西安交通大学出版社
地　　址	西安市兴庆南路25号(邮编：710049)
网　　址	http://unit.xjtu.edu.cn/unit/jtupress
电　　话	(029)82668357　82667874(发行部) (029)82668315　82669096(总编办)
电子信箱	eibooks@163.com
印　　刷	西安新视点印务有限责任公司
版　　次	2003年12月第1版　2005年8月第2次印刷
开　　本	787mm×1 092mm　1/16
印　　张	12.75
字　　数	300千字
书　　号	ISBN 7-5605-1783-8/TP·359
定　　价	18.00元

21世纪高职高专计算机系列教材编委会

重印说明

自2002年底以来，我社已陆续出版“21世纪高职高专计算机系列教材”18种。这批教材出版以来，受到全国众多高职高专院校师生的欢迎，许多教材出版仅数月就被重印，且需求量有迅速增长之势。这使我们深感欣慰，并深受鼓舞。

为满足广大高职高专院校师生对高水平教材的要求，我们在每次重印教材前都收集并采纳了该教材作者们以及许多使用该教材的教师、学生的反馈意见和建议，特别是勘误信息，对原教材进行订正，使之不断完善。我们也期待更多的读者给我们反馈教材修订意见和建议（我们的联系方式请参看本书版权页）。

随着高职高专计算机教育的不断发展，原定出版的19种教材远不能满足需要。我们将在充分调研并与广大师生诚挚合作的基础上，继续推出更多适应高职高专教学需要的优秀教材。目前，已有2种新教材列入本系列教材的出版计划，从而使本系列教材的品种增至21种。今后这一数字将会随着新教材的加入而不断更新。更新后的本系列教材目录将展现在每次重印或修订后教材的封底中，请读者给予关注。此外，读者可以通过来函或电子邮件索取最新教材目录以获取相关信息。

我们真诚地期待广大读者，特别是使用本系列教材的教师和学生，继续支持和关注我们的教材出版工作，对我们工作中的不足之处提出坦率的批评和中肯的建议；我们也期待与更多的优秀教师合作，出版更多更优秀的教材，为我国高等教育事业的发展贡献绵薄之力。

西安交通大学出版社

计算机图书编辑室

2005年8月

序

随着我国科学技术的发展，全民高等教育已经成为时代的要求。扩大招生规模，发展高等职业教育，已经成为各级政府和广大教育工作者的共识。为了指导和推动全国高等职业教育的健康发展，教育部先后制定了“高职高专教育基础课程教学基本要求”和“高职高专教育专业人才培养目标和规格”两个文件。在此基础上，许多出版社先后出版了相关的系列教材，对推动我国的高等职业教育起到了积极的作用。

时代在前进，科学技术在发展，尤其是计算机信息技术发展的速度更是惊人，这就要求高等学校的教学内容应能跟上科学技术的发展，应能满足新技术对新型人才的需求；教材应当不断地修改和更新。故此，我们组织高校中长期从事高等职业教育的专家、学者编写了“21 世纪高职高专计算机系列教材”。在编写过程中，我们以教育部的上述两个文件为依据，参阅同类教材，汲取多年来在高等专科教育、成人教育中培养应用型人才的成功经验，充分体现高职高专实用型人才的特征，“以应用为目的，以必须、够用为度”，尽量做到从实际应用的需求出发，减少枯燥乏味的纯理论和概念，使学生理论联系实际，学中有用，边学边用，通过学习提高应用和解决实际问题的能力。在编排顺序方面，尽量做到由浅入深，循序渐进，内容多样，结构合理，语言简练，文字流畅，使学生易学、易懂、易掌握。

这套教材目前已列入选题的有 19 种，既有专业基础知识，又有最新技术，可作为高职高专基础课、专业基础课以及最新技术课的教材，也可供自考和学历文凭教育使用。

在 21 世纪到来的时候，我国高等职业教育迅猛发展的格局已经形成。这就要求教育界的志士仁人奋发努力，以自己的心血和汗水去培养时代所需要的一代有理想、有道德、有知识、有能力的高素质、高水平的应用型专业人才。

陈建铎

2002 年 10 月

前　言

“数据结构”是计算机专业的核心课程，它为计算机软件开发、应用人员提供重要的专业基础知识和必要的技能训练。

现实中许许多多的复杂问题都可上升为数学模型，人们通过对数学模型的分析，确定解决问题的步骤——算法。数据结构作为一个专业术语，仅仅是指一种数学模型，这种模型体现了数据的组织形式和基本操作方式。“数据结构”课程脱胎于“离散数学结构”，它涉及的数学模型为各种离散结构（如向量、集合、树、图等）。学习“数据结构”课程的目的在于学会分析计算机加工的数据的特性，以便为应用涉及的数据选择适当的逻辑结构、存储结构以及实现各种操作的算法，并初步掌握算法的时间分析和空间分析技术。

用计算机解决任何问题都离不开程序设计，程序设计的实质是数据的表示和处理。本课程的学习过程也是复杂程序设计的训练过程。

本书内容共分为 8 章。第 1 章介绍数据结构的一般概念和算法分析的初步知识；第 2～6 章分别介绍线性表、栈和队列、串和数组、树与二叉树、图等数据结构的存储结构和不同存储结构上的一些操作的实现算法；第 7 章讨论了各种查找表及查找方法；第 8 章讨论了各种排序算法。书中打星号部分为选学内容。本书计划学时为 70 学时左右，其中上机实习为 20 学时。

本书是为计算机类高职高专学生编写的教材。针对高职高专面向应用、注重实践的特点，本书力求做到选材精练、叙述简洁，尽量避免抽象理论的介绍和复杂公式的推导。对各种数据结构均从实际出发，通过对实例的分析，使学生理解数据结构的基本概念。书中安排多个综合实例和完整的 C 语言程序供学生上机实习参考，每章带有小结和适量的习题。考虑到专升本考试和其他考试的需要，在习题中编排了较多的选择题和填空题。学习本书需要有 C 语言程序设计的知识，同时最好还具有离散数学中的集合论和图论的知识。

本书第 1，4 章由周筱媛编写，第 2 章由刘文哲编写，第 3 章由王敏编写，第 5，8 章由王忠义编写，第 6，7 章由张小艳编写，王忠义担任主编。

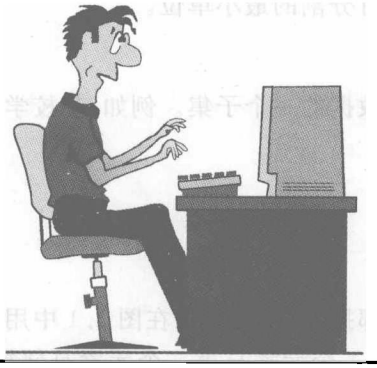
西安科技大学计算机系龚尚福教授审阅了本书全稿，提出了宝贵的意见和建议，在此表示衷心的感谢。

由于作者水平所限，书中难免会有不足和错误之处，欢迎读者批评指正。

编　者

2003 年 8 月

目　录



第1章 概 述

用数字计算机解决任何问题都离不开程序设计，程序设计的实质是对数据的处理。在程序设计中会涉及到各种各样的数据，为了对数据进行加工处理，需要讨论数据之间的关系，需要存储和组织数据。在进行程序设计与开发的过程中，不仅要描述处理数据的算法，还应对实现同一操作的不同算法进行比较，以选择时间和空间性能较好的算法。本章介绍数据、数据元素等术语，讨论数据结构、算法等基本概念，并简单介绍描述算法的语言和算法性能分析的初步知识。

1.1 数据结构的基本概念

1.1.1 基本术语

在对数据结构下定义之前，让我们先对全书中使用的名词和术语赋以确定的含义。

1. 数据

数据是指那些能被输入到计算机中，且被计算机处理的各种符号所组成的集合，是对客观事物的符号表示，是计算机操作的对象的总称。通俗地讲，数据就是信息的载体，是计算机程序加工的“原料”。例如，在数值计算中使用的数据是整数和实数；在一个编译程序或文本编辑程序中使用的数据是字符串。随着计算机技术的发展，如图像和声音等信息载体都可以经过一定的转换变成数字化的信息数据，以便利用计算机进行加工和处理。因此，对计算机科学而言，数据的含义极为广泛。

2. 数据元素

数据元素是数据中的一个“个体”，是数据的基本单位。在数据处理中通常作为一个整体来考虑和处理。例如，在学籍管理中记录每个学生的数据就是一个数据元素。通常一个数据元素由一个或若干个数据项组成。有时也把数据元素称为结点或记录。

3. 数据项

数据项是数据结构中讨论的最小单位，具有独立的含义。例如，学生信息中的每一项(如

学号、姓名、性别等)为一个数据项。数据项是数据的不可分割的最小单位。

4. 数据对象

数据对象是具有相同特性的数据元素的集合,它是数据的一个子集。例如,某校学籍管理中的数据对象就是该校的全体学生记录。

1.1.2 数据结构与数据类型

1. 数据的逻辑关系与逻辑结构

所谓逻辑关系是指数据元素之间的关联方式或称"邻接关系"。例如在图 1.1 中用小圆圈表示数据元素,用连线表示元素之间的邻接关系,左边第一个元素与第二个元素是邻接的,即相互关联的,因此这两个元素之间有逻辑关系。但是第一个元素与第三个元素不邻接,即相互不直接关联,因此它们之间没有逻辑关系。数据元素之间逻辑关系的整体称为逻辑结构。"结构"二字可理解为数据元素之间的关系。这里的"邻接关系"是一种抽象的关系,对于各种实际问题,它可以代表不同的关系。

数据的逻辑结构可大致分为线性结构和非线性结构。线性结构的逻辑特征是有且仅有一个开始元素和一个终止元素,除第一个和最后一个元素之外,其余元素有且仅有一个直接前趋和直接后继,如图 1.1 所示。

非线性结构的逻辑特征是一个结点可能有零个或多个直接前趋或直接后继,包括树形结构、图状结构和集合,如图 1.2 所示。

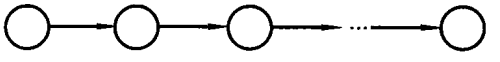


图 1.1 线性结构

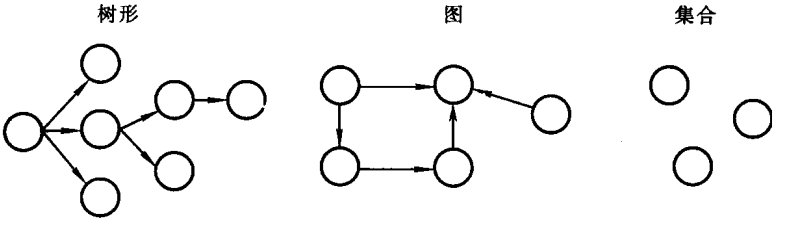


图 1.2 非线性结构

数据的逻辑结构属于什么类型,与数据元素本身的形式、内容无关,与数据元素的相对位置无关,与数据元素的个数无关。一些表面上很不相同的数据可以有相同的逻辑结构。因此,逻辑结构是数据组织的某种"本质性"的东西,是数据组织的主要方面。

2. 数据的存储结构

数据的存储结构即数据在计算机存储器中具体存放的物理形式,包括数据元素和元素间的关系在计算机内的表示。在计算机中可以用一个由若干位组合起来的位串表示一个数据元素。由于存储方式的不同,数据元素之间的关系有两种不同的表示方法——顺序映像和非顺序映像,由此得到两种不同的存储结构:顺序存储结构(Sequential Storage)和链式存储结构(Linked Storage Structure)。顺序存储结构的特点是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。因此,在这种存储结构中可以通过一个简单、直接的公式随机存取

任意元素，但其缺点是需要较大的存储空间且空间利用率低，插入和删除操作需移动大量的元素。链式存储结构的特点是以附加信息（指针）来表示数据元素之间的关系。因此，在这种结构中，克服了顺序存储结构的缺点，但存取数据元素时必须从第一个元素开始沿指针依次找到要访问的元素。

同一种逻辑结构可以选择不同的存储结构来表示。经过后续章节的学习会看到，我们常常把同一种逻辑结构的不同存储结构用不同的名字来标识。例如，具有线性结构的线性表，若用顺序存储方式存储时，称为顺序表，而用链接方式存储时，称为链表。

3. 数据的操作

数据的操作也称为数据的运算。对数据的操作的种类是没有限制的，可根据需要来定义。数据结构课程中讨论较多的操作有以下几种：

（1）插入　在带有结构的数据集合中的指定位置上添加新的数据元素；

（2）删除　在带有结构的数据集合中删除某个指定的数据元素；

（3）更新　在带有结构的数据集合中改变某个数据元素的值，在概念上等价于删除和插入操作的组合；

（4）查找　在带有结构的数据集合中寻找满足某个特定要求的数据元素（的位置和值）；

（5）排序　（在线性结构中）重新安排数据元素之间的逻辑顺序关系，使其按值由小到大或由大到小的次序排列。

从操作的特性上来分，所有的操作可归结为：加工型操作（操作改变了操作之前的结构的“值”，如结点个数、某些结点的内容、结点之间的关系）和引用型操作（操作不改变结构的值，只是查询或求得结构的值）。不难看出，上述5种操作中“查找”为引用型操作，其余均为加工型操作。

4. 数据结构的定义

数据结构作为一个专业术语，是指相互之间存在一种或多种特定关系的数据元素的集合，可以用一个二元组来表示，即

数据结构 = <D,R>

其中，D是数据元素的有限集，R是D上关系的有限集。

数据结构的定义，可简述为“带有结构的数据集合”。这里对数据结构的定义，与前面数据的逻辑结构的定义实质上是一样的。

但是，数据的逻辑结构、存储结构和数据的操作三者之间关系密切，为了对数据进行处理，必须把它们作为整体来研究。所以，数据结构作为一门课程，它不仅研究数据的逻辑结构，而且研究数据的存储结构以及在不同存储结构上实现各种操作的算法。

习惯上，人们针对数据的逻辑结构以及基本操作的特点，冠以不同的名字，称为不同的数据结构（实质上是下节介绍的抽象数据类型）。例如，当数据的逻辑结构为线性结构，基本操作为前面介绍的插入、删除、更新等操作时称为线性表；同样是线性结构，当操作受到一定限制时，譬如，当插入和删除操作只能在数据表的一端进行时，则称为堆栈。

5. 数据类型

在用高级程序语言编写程序时，必须对程序中出现的每个变量、常量或表达式，明确说明它们所属的数据类型。因为类型明显或隐含地规定了在程序执行期间，变量或表达式所有可

能取值的范围，以及它们所允许进行的操作。因此，数据类型是一个值的集合和定义在此集合上的一组操作的总称，它是数据结构在程序设计语言中的实现。

按“值”是否可分解，高级语言中的数据类型可分为两类：一类是非结构的原子类型，原子类型的值是不可分解的，如 C 语言中的整型、字符型等标准类型及指针等简单的导出类型；另一类是结构类型，其值可分解为若干成分，如 C 语言中的数组、结构体等类型。原子类型通常是由语言直接提供的，而结构类型则是用户借助语言提供的描述机制自己定义的，由标准类型派生的。

实际上，在计算机中每个处理器都提供了一组原子类型或结构类型，其操作是通过计算机设计的一套指令系统直接由电路系统完成的，而高级语言中的数据类型是通过编译器或解释器转化成低级语言的数据类型实现的。使用数据类型，对程序设计者来讲不需要了解其实现过程的一切细节，从而将其注意力集中在该数据类型及操作的抽象特性上。

如果再提高数据类型的抽象级别，使其应用的范围更广，就形成了抽象数据类型。抽象数据类型不仅包括数据的逻辑结构，而且还包括定义在该逻辑结构上的基本操作集。下一节将介绍抽象数据类型的概念。

* 1.2 抽象数据类型

抽象数据类型，简称 ADT(Abstract Data Type)，是指一个数学模型以及定义在此数学模型上的一组操作。它可看作是数据的逻辑结构及其在逻辑结构上定义的操作。

抽象数据类型可用(D,R,P)三元组表示，其中，D 是数据对象，R 是 D 上的关系集，P 是对 D 的基本操作集。那么一个 ADT 可表示为：

```
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
```

其中，数据对象和数据关系的定义用伪码描述，而基本操作的定义格式为：

```
基本操作名(参数表)
初始条件：<初始条件描述>
操作结果：<操作结果描述>
```

在这里基本操作有两种参数，一种是赋值参数，它只为操作提供输入值；另一种是传递地址参数，以 & 打头，它除了提供输入值外，还将返回操作结果。

“初始条件”描述了操作执行之前数据结构和参数应满足的条件，若不满足，则操作失败，并返回相应的出错信息。

“操作结果”说明了操作正常完成之后，数据结构的变化状况和应返回的结果。若初始条件为空，则省略之。

例 1.1 抽象数据类型三元组的定义。

```
ADT Triplet{
    数据对象：D = {e1,e2,e3 | e1,e2,e3∈ElemSet (定义了关系运算的某个集合) }
```

数据关系:R1 = {<e1,e2>,<e2,e3>}

基本操作:

InitTriplet(&T,v1,v2,v3)

操作结果:初始化三元组 T,元素 e1,e2,e3 分别被赋以参数 v1,v2,v3 的值。

DestroyTriplet(&T)

操作结果:三元组 T 被销毁。

Get(T,i,&x)

初始条件:三元组 T 已存在,1<=i<=3。

操作结果:用 x 返回 T 的第 i 个元素的值。

Put(&T,i,x)

初始条件:三元组 T 已存在,1<=i<=3。

操作结果:改变 T 的第 i 个元素的值为 x 。

IsAscending(T)

初始条件:三元组 T 已存在。

操作结果:如果 T 的三个元素按升序排列,则返回 1,否则返回 0。

IsDescending(T)

初始条件:三元组 T 已存在。

操作结果:如果 T 的三个元素按降序排列,则返回 1,否则返回 0。

Max(T,&x)

初始条件:三元组 T 已存在。

操作结果:用 x 返回 T 的三个元素中的最大值。

Min(T,&x)

初始条件:三元组 T 已存在。

操作结果:用 x 返回 T 的三个元素中的最小值。

}ADT Triplet

抽象数据类型可以看作是描述问题的模型,独立于具体的实现。它有两个重要的优点:数据抽象和数据封装。用 ADT 描述程序处理的实体时,强调的是其本质的特征、其所能完成的功能以及它和外部用户的接口(即外界使用它的方法),即数据抽象。同时,ADT 将数据和操作封装在一起,使得用户程序只能通过在 ADT 里定义的操作来访问其中的数据,从而实现信息隐藏。在 C++中可以用类的说明来表示 ADT,用类的实例来实现 ADT。

抽象数据类型的定义由一个值域和定义在该值域上的一组操作组成。按其值的不同特性,可细分为下列三种类型:

- 原子类型(Atomic Data Type) 属原子类型的变量的值是不可分解的。
- 固定聚合类型(Fixed-Aggregate Data Type) 属该类型的变量,其值由确定数目的成分按某种结构组成。
- 可变聚合类型(Variable-Aggregate Data Type) 和固定聚合类型相比较,构成可变聚合类型"值"的成分数目不确定。

显然,后两种类型可统称为结构类型。

本书采用 C 语言描述算法,C 语言不能整体描述抽象数据类型,为简单起见,本书后续章

节中未采用伪代码描述抽象数据类型，而是将数据的各种逻辑结构和其基本操作分开来介绍。

1.3 算法及算法描述

因为对数据的操作是通过算法(Algorithm)描述的，所以讨论算法是数据结构课程的重要内容之一。

1.3.1 算法的定义和性质

算法(Algorithm)是对特定问题求解步骤的一种描述，它是指令的有限序列，一条指令表示一个或多个操作。一个算法必须满足以下 5 个重要特性。

1. 有穷性

对于任意一组合法输入值，在执行有穷步骤之后一定能结束，算法中的每个步骤都能在有限时间内完成。

2. 确定性

对于每种情况下所应执行的操作，在算法中都有确切的规定，使算法的执行者或阅读者都能明确其含义以及如何执行，不能出现歧义性。例如，若算法中出现“当 a 比 b 大很多时，就…”，这里的“大很多”就存在不确定性。

3. 可行性

算法中的每一步都必须是足够基本的。就是说，它们原则上都能精确地执行。

4. 输入

作为算法加工对象的量值，通常体现为算法中的一组变量。有些输入量需要在算法执行过程中输入，而有的算法表面上可以没有输入，实际上要处理的数据已被嵌入算法之中。

5. 输出

它是一组与“输入”有确定关系的量值，是算法进行信息加工后得到的结果，这种确定关系即为算法的功能。

1.3.2 算法的评价

求解同一问题时可能有许多不同的算法，究竟如何来评价这些算法的优劣，以便从中选出较好的算法呢？

在设计或评价一个算法时，通常应考虑达到以下目标。

1. 正确性

首先，算法应当满足以特定的“规格说明”的方式给出的需求。其次，对算法是否“正确”的理解可以有以下 4 个层次：

(1) 程序中不含语法错误；

(2) 程序对于几组输入数据能够得出满足要求的结果；

(3) 程序对于精心选择的、典型的、苛刻且带有刁难性的几组输入数据能够得出满足要求