



高等职业教育
计算机类课程规划教材

数据结构 实训与习题解析

GAODENG ZHIYE JIAOYU
JISUANJILEI KECHENG GUIHUA JIAOCAI

新世纪高等职业教育教材编审委员会组编

主编 刘俞 景秀丽 主审 黄建华

大连理工大学出版社



新世纪

高等职业教育计算机类课程规划教材

数 据 结 构

实训与习题解析

新世纪高等职业教育教材编审委员会组编

主审 黄建华

主 编 刘 俞 景秀丽 副主编 安训国 汤 凡

SHUJU JIEGOU SHIXUN YU XITI JIEXI

大连理工大学出版社

DALIAN UNIVERSITY OF TECHNOLOGY PRESS

© 大连理工大学出版社 2006

图书在版编目(CIP)数据

数据结构实训与习题解析 / 刘俞, 景秀丽主编. — 大连: 大连理工大学出版社, 2006. 4

高等职业教育计算机类课程规划教材

ISBN 7-5611-3164-X

I. 数… II. ①刘… ②景… III. 数据结构—高等学校: 技术学校—教学参考资料 IV. TP311.12

中国版本图书馆 CIP 数据核字(2006)第 034289 号

大连理工大学出版社出版

大连市软件园路 80 号 邮政编码 116023

发行: 0411-84708842 邮购: 0411-84703636 传真: 0411-84701466

E-mail: dutp@ dutp. cn URL: <http://www. dutp. cn>

大连理工印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸: 185mm × 260mm 印张: 15.5 字数: 340 千字

印数: 1 ~ 4 000

2006 年 4 月第 1 版

2006 年 4 月第 1 次印刷

责任编辑: 潘弘喆

责任校对: 董 静

封面设计: 波 朗

定 价: 24.00 元

新世纪高等职业教育教材编委会教材建设 指导委员会

主任委员:

曹勇安 黑龙江东亚学团董事长 齐齐哈尔职业学院院长 教授

副主任委员(以姓氏笔画为序):

马必学	武汉职业技术学院院长	教授
王大任	辽阳职业技术学院院长	教授
冯伟国	上海商业职业技术学院副院长	教授 博士
刘兰明	邯郸职业技术学院副院长 教授	博士
刘长声	天津对外经济贸易职业学院副院长	副教授
李竹林	河北建材职业技术学院院长	教授
李长禄	黑龙江工商职业技术学院副院长	副研究员
陈 礼	广东顺德职业技术学院副院长	教授
金长义	广西工业职业技术学院院长	副教授
赵居礼	陕西工业职业技术学院副院长	副教授
徐晓平	盘锦职业技术学院院长	教授
高树德	吉林交通职业技术学院副院长	教授
戴裕崴	天津轻工业职业技术学院副院长	副研究员 博士

秘书长:

杨建才 沈阳师范大学职业技术学院院长

副秘书长(以姓氏笔画为序):

张和平 江汉大学高等职业技术学院院长
张化疆 黑龙江生态工程职业学院副院长
周 强 齐齐哈尔大学应用技术学院院长

秘书组成员(以姓氏笔画为序):

卜 军 上海商业职业技术学院
王澄宇 大庆职业学院
粟景妆 广西国际商务职业技术学院
鲁 捷 沈阳师范大学职业技术学院
谢振江 黑龙江省司法警官职业学院

会员单位(略):

总 序

我们已经进入了一个新的充满机遇与挑战的时代,我们已经跨入了 21 世纪的门槛。

20 世纪与 21 世纪之交的中国,高等教育体制正经历着一场缓慢而深刻的革命,我们正在对传统的普通高等教育理论教学与社会发展的现实需要不相适应的现状作历史性的反思与变革的尝试。

20 世纪最后的几年里,高等职业教育的迅速崛起,是影响高等教育体制变革的一件大事。在短短的几年时间里,普通中专教育、普通高专教育全面转轨,以高等职业教育为主导的各种形式的培养应用型人才的教育发展到与普通高等教育等量齐观的地步,其来势之迅猛,迫人深思。

无论是正在缓慢变革着的普通高等教育,还是迅速推进着的培养应用型人才的高等职业教育,都向我们提出了一个同样的严肃问题:中国的高等教育为谁服务,是为教育发展自身,还是为包括教育在内的大千社会?答案肯定而且惟一,那就是教育也置身其中的现实社会。

由此又引出高等教育的目的问题。既然教育必须服务于社会,它就必须按照不同领域的社会需要来完成自己的教育过程。换言之,教育资源必须按照社会划分的各个专业(行业)领域(岗位群)的需要实施配置,这就是我们长期以来明乎其理而疏于力行的学以致用问题,这就是我们长期以来未能给予足够关注的教育的目的问题。

如所周知,整个社会由其发展所需要的不同部门构成,包括公共管理部门如国家机构、基础建设部门如教育研究机构和各种实业部门如工业部门、商业部门,等等。每一个部门又可作更为具体的划分,直至同它所需要的各种专门人才相对应。教育如果不能按照实际需要完成各种专门人才培养的目标,就不能很好地完成社会分工所赋予它的使命,而教育作为社会分工的一种独立存在就应受到质疑(在市场经济条件下尤其如此)。可以断言,按照社会的各种不同需要培养各种直接有用人才,是教育体制变革的终极目的。

随着教育体制变革的进一步深入,高等院校的设置是否会同社会对人才类型的不同需要一一对应,我们姑且不论。但高等教育走应用型人才培养的道路和走理论型(也是一种特殊应用)人才培养的道路,学生们根据自己的偏好各取所需,始终是一个理性运行的社会状态下高等教育正常发展的途径。

高等职业教育的崛起,既是高等教育体制变革的结果,也是高等教育体制变革的一个阶段性表征。它的进一步发展,必将极大地推进中国教育体制变革的进程。作为一种应用型人才培养的教育,高等职业教育从专科层次起步,进而高职本科教育、高职硕士教育、高职博士教育……当应用型人才培养的渠道贯通之时,也许就是我们迎接中国教育体制变革的成功之日。从这一意义上说,高等职业教育的崛起,正是在为必然会取得最后成功的教育体制变革奠基。

高职教育还刚刚开始自己发展道路的探索过程,它要全面达到应用型人才培养的正常理性发展状态,直至可以和现存的(同时也正处在变革分化过程中的)理论型人才培养的教育并驾齐驱,还需假以时日;还需要政府教育主管部门的大力推进,需要人才需求市场的进一步完善发育,尤其需要高职教学单位及其直接相关部门肯于做长期的坚忍不拔的努力。新世纪高等职业教育教材编审委员会就是由北方地区100余所高职院校和出版单位组成的旨在以推动高职教材建设来推进高等职业教育这一变革过程的联盟共同体。

在宏观层面上,这个联盟始终会以推动高职教材的特色建设为己任,始终会从高职教学单位实际教学需要出发,以其对高职教育发展的前瞻性的总体把握,以其纵览全国高职教材市场需求的广阔视野,以其创新的理念与创新的组织形式,通过不断深化的教材建设过程,总结高职教学成果,探索高职教材建设规律。

在微观层面上,我们将充分依托众多高职院校联盟的互补优势和丰裕的人才资源优势,从每一个专业领域、每一种教材入手,突破传统的片面追求理论体系严整性的意识限制,努力凸现高职教育职业能力培养的本质特征,在不断构建特色教材建设体系的过程中,逐步形成自己的品牌优势。

新世纪高等职业教育教材编审委员会在推进高职教材建设事业的过程中,始终得到了各级教育主管部门(如国家教育部、辽宁省教育厅)以及各相关院校相关部门的热忱支持和积极参与,对此我们谨致深深谢意;也希望一切关注、参与高职教育发展的同道朋友,在共同推动高职教育发展、进而推动高等教育体制变革的进程中,和我们携手并肩,共同担负起这一具有开拓性挑战意义的历史重任。

新世纪高等职业教育教材编审委员会

2001年8月18日

《数据结构实训与习题解析》是新世纪高职教材编委会组编的高等职业教育计算机类课程规划教材之一,同时也是《数据结构》(第三版)的配套教材。

数据结构是计算机专业的核心课程,是计算机程序设计的重要理论技术基础。这是一门较为复杂和难以理解的课程,在传统的教学中,通常只注重课堂教学,而忽视了实践环节。学生对所学的知识无法深刻理解,或学会了也不知道有什么用的情况比较普遍,造成了学生对这门课程的兴趣不高,因此,教学效果难以令人满意。为了提高学生的学习兴趣及训练学生将数据结构中的知识用于复杂程序设计的技能,必须注重实训环节,使学生学了就去用,在设计过程中自己发现问题,利用学到的知识解决问题,加深理解所学知识的本质,使其在实践中真正掌握各个知识点。经过不断的实践训练,使学生充分掌握各数据结构及其算法,提高程序设计技巧,提高分析问题、解决问题的能力。

《数据结构实训与习题解析》以 C 语言为开发工具,以加强学生对数据结构算法实际应用技能的训练,提高分析问题和解决问题的能力为目的。本书选取并讨论了多种应用的项目设计,以项目开发的形式提出问题,分析问题,最终解决问题。其中选取的项目案例贴近实际生活,能充分调动学生的学习积极性,并易于理解和接受。

全书共分 8 章,每章都分为:基础知识篇、实训篇、实训项目、基本题和习题解析。按数据结构教材内容的先后顺序给出了相关的实训项目,其中包括线性表、栈和队列、串、树、图以及查找和排序。实训项目既涵盖各知识点,又贴近学习和实际生活。书中对各项目的设计思路、方法进行了较详细的分析。并给出主教材中全部习题的解析和参考答案。全书特别注重对实际应用问题的分析和理解,内容丰富,概念清楚,思想严谨,方法科学。

本书中对所有实训项目的实现都给出了完整的 C 语言程序,对关键算法及语句都给出了详细的注释,并给出程序的运行结果。所有程序都在 Turbo C 2.0 环境下调试通



过。

本教材由刘俞、景秀丽任主编,安训国、汤凡任副主编。第1章绪论、第2章线性表、第5章树、第7章查找、第8章排序的实训部分由刘俞编写,第3章栈和队列、第4章串、第6章图的实训部分由景秀丽编写。安训国和汤凡给出了《数据结构》一书完整的习题解析。刘俞负责全书的统稿。黄建华审阅了全部书稿,并提出宝贵的意见。

尽管我们在本教材中对数据结构课程的实践环节做了大量的探索和尝试,但由于我们的水平有限、时间仓促、书中难免存在一些错误和不合理之处,恳请广大读者批评指正。

所有意见、建议请寄往:gzjckfb@163.com

联系电话:0411-84707492 84706104

编者

2006年4月



第1章 绪论	1
1.1 基础知识篇	1
1.1.1 基本概念和术语	1
1.1.2 数据的逻辑结构	2
1.1.3 数据的存储结构	2
1.1.4 算法的描述	2
1.2 实训篇	3
1.2.1 实训步骤	3
1.2.2 C语言环境中的动态调试法	4
1.2.3 常用的调试途径	5
1.2.4 实训报告的内容与书写格式	6
1.3 习题解析	6
第2章 线性表	10
2.1 基础知识篇	10
2.1.1 基本概念	10
2.1.2 线性表的顺序存储及运算实现	11
2.1.3 线性表的链式存储及运算实现	14
2.1.4 顺序表和链表的比较	17
2.2 实训篇	18
2.2.1 超市密码存储箱系统的设计与实现	18
2.2.2 员工通讯录管理系统	25
2.2.3 约瑟夫环问题	32
2.2.4 模拟蜗牛爬树	35
2.3 实训项目	40
2.4 基本题	41
2.4.1 选择题	41
2.4.2 填空题	41
2.5 习题解析	44
第3章 栈和队列	54
3.1 基础知识篇	54
3.1.1 栈	54
3.1.2 队列	59

3.2 实训篇	65
3.2.1 栈的应用——表达式求值问题	66
3.2.2 队列的应用——迷宫问题	71
3.3 实训项目	75
3.4 基本题	76
3.4.1 选择题	76
3.4.2 填空题	76
3.5 习题解析	78
第4章 串	86
4.1 基础知识篇	86
4.1.1 串及其存储结构	86
4.1.2 串的基本运算	89
4.2 实训篇	92
4.3 实训项目	94
4.4 基本题	95
4.4.1 选择题	95
4.4.2 填空题	95
4.5 习题解析	96
第5章 树	102
5.1 基础知识篇	102
5.1.1 树的基本概念	102
5.1.2 二叉树	102
5.1.3 树的存储结构	106
5.1.4 哈夫曼树	107
5.2 实训篇	108
5.2.1 二叉树的建立及三种遍历的实现	108
5.2.2 哈夫曼编码/译码系统	116
5.2.3 六皇后问题求解及演示	122
5.3 实训项目	125
5.4 基本题	125
5.4.1 选择题	125
5.4.2 填空题	126
5.5 习题解析	127
第6章 图	140
6.1 基础知识篇	140
6.1.1 图的基本概念	140
6.1.2 图的存储结构	143
6.1.3 图的基本操作	145

6.2 实训篇	153
6.2.1 图的遍历问题	154
6.2.2 所有顶点对最短路径问题	156
6.3 实训项目	160
6.4 基本题	161
6.4.1 选择题	161
6.4.2 填空题	163
6.5 习题解析	164
第7章 查找	170
7.1 基础知识篇	170
7.1.1 基本概念	170
7.1.2 静态查找	170
7.1.3 动态查找	172
7.2 实训篇	175
7.2.1 手机短信内容中电话号码和手机号码的识别与提取	176
7.2.2 数据结构教材中各功能函数的存放与查找	180
7.2.3 二叉排序树的使用	184
7.3 实训项目	188
7.4 基本题	188
7.4.1 选择题	188
7.4.2 填空题	189
7.5 习题解析	190
第8章 排序	199
8.1 基础知识篇	199
8.1.1 排序的基本概念	199
8.1.2 插入排序	200
8.1.3 交换排序	201
8.1.4 选择排序	202
8.1.5 归并排序	204
8.1.6 基数排序	204
8.1.7 内部排序法的比较和选择	205
8.2 实训篇	205
8.2.1 药店的药品销售统计系统	205
8.2.2 学生成绩排名	214
8.2.3 电视大赛观众投票及排名系统	218
8.3 实训项目	221
8.4 基本题	222
8.4.1 选择题	222

10 / 数据结构实训与习题解析 □

8.4.2 填空题	222
8.5 习题解析	223
附录 基本题参考答案	232
参考文献	235

第 1 章

绪 论

1.1 基础知识篇

学习要点:

- 数据、数据结构、算法的基本概念
- 算法评价:时间复杂度、空间复杂度的概念
- 程序设计的步骤
- 程序的调试方法
- 程序的测试方法
- 实验报告的内容及格式

1.1.1 基本概念和术语

1. 数据(Data)

数据是对客观事物的符号表示。在计算机科学中其含义是指所有能够输入到计算机中并被计算机程序处理的符号集合。

2. 数据元素(Data Element)

数据元素是数据集合中的一个实体,是计算机程序加工处理的基本单位。数据元素按其组成可分为简单型数据元素和复杂型数据元素。简单型数据元素由一个数据项组成,所谓数据项就是数据中不可再分割的最小单位;复杂型数据元素由多个数据项组成。

3. 数据类型(Data Type)

数据类型是一组性质相同的值的集合以及定义于这个集合上的一组操作的总称。

4. 数据对象(Data Object)

数据对象是性质相同的数据元素组成的集合,是数据的一个子集。例如,整数数据对象的集合可表示为 $N = \{0, \pm 1, \pm 2, \dots\}$, 字母字符数据对象的集合可表示为 $C = \{'A', 'B', \dots, 'Z'\}$ 。

5. 数据结构(Data Structure)

数据结构的基本含义是指数据元素之间的关系,换句话说,数据元素之间的相互关系称为结构,带有结构的数据对象称为数据结构。

1.1.2 数据的逻辑结构

数据结构中所说的“关系”实际上是指数据元素之间的逻辑关系,又称为逻辑结构。常见的数据逻辑结构有:集合、线性结构、树形结构和图形结构。

1.1.3 数据的存储结构

数据的存储结构(或称为物理结构)是指数据结构在计算机存储器中的具体实现。

常见的存储结构:顺序存储结构——借助于数据元素的相对存储位置来表示数据元素之间的逻辑结构;链式存储结构——借助于指示数据元素地址的指针来表示数据元素之间的逻辑结构。

1.1.4 算法的描述

1. 算法的概念

算法是解决某个特定问题的一种方法或一个有限过程。

计算机对数据的操作可以分为数值型和非数值型两种类型。在数值型操作中主要进行的是算术运算;而在非数值型操作中主要进行的是检索、排序、插入、删除等等。

2. 设计算法的基本过程

通过对问题进行详细地分析,抽象出相应的数学模型;

确定使用的数据结构,并在此基础上设计各种操作的算法;

选用某种语言将算法转换成程序;

调试并运行这些程序。

3. 算法应该具有下列五个特性

(1)有穷性:一个算法必须在执行有穷步之后结束。

(2)确定性:算法中的每一步,必须有确切的含义,不会产生二义性。

(3)可行性:算法中描述的每一步操作都可以通过已有的基本操作执行有限次实现。

(4)输入:一个算法应该有零个或多个输入。

(5)输出:一个算法应该有一个或多个输出。这里所说的输出是指与输入有某种特定关系的量。

4. 算法的设计要求

(1)正确性:要求算法能够正确地执行预先规定的功能,并达到所期望的性能要求。

(2)可读性:为了便于理解、测试和修改,算法应该具有良好的可读性。

(3)健壮性:算法中拥有对输入数据、打开文件、读取文件记录、分配内存空间等操作的结果检测,并通过与用户对话的形式做出相应的处理选择。

(4)高效性:算法的效率是指算法执行的时间和占用的存储空间。如果对于一个问题有多个算法可以实现,应尽可能选择执行时间短、占用空间少的算法。

5. 算法的评价

评价一个算法优劣的主要指标是时间效率和空间效率。

(1) 算法的时间效率

算法的时间效率是指算法运行时所需时间的多少。算法的时间效率主要由两个因素决定:所需处理问题的数据量大小;在解决问题的过程中,基本操作的执行次数。

时间效率的分析:如果我们将一个算法所花费的时间设计成一个以数据量 n 为自变量的函数 $T(n)$,这个函数在正整数定义域范围内一定是单调递增的。好的算法应该能够在数据量 n 增长时,使函数 $T(n)$ 的值以比较缓慢的速度增长。

(2) 算法的空间效率

算法的空间效率是指在算法的执行过程中,所占据辅助空间的大小。辅助空间就是除算法代码本身和输入输出数据所占据的空间外,算法临时开辟的存储空间。在有些算法中,占据辅助空间的大小与所处理的数据量有关,而有些却与其无关。后一种是较理想的情况。在设计算法时,应该注意空间效率。

1.2 实训篇

1.2.1 实训步骤

程序设计应该遵循软件开发的一般规律和步骤。其主要步骤如下:

1. 问题分析

正确地理解问题,分析究竟要“做什么”,分析问题已有的信息及其作用,分析在解决问题中对信息处理的规则、要求及限制条件,分析解决问题后应该输出什么样的结果(输出的形式、格式等),并分析得出判定结果是否正确的标准。

2. 设计分析

得出解决问题的思路、主要流程、采用的数据结构类型的说明、主要算法的思想。

3. 设计方案

采用的数据结构类型的定义、主要算法的描述及说明。

4. 编写程序

根据问题的要求和已得到的算法编写程序。

5. 调试程序

目的是发现程序中的语法错误及一些逻辑错误(还有一部分逻辑错误在测试中才能被发现),并修改,使程序能够运行。可以采用 C 语言环境提供的动态调试法。

6. 测试程序

选择若干具有代表性的输入数据,其中包括合理的数据(包括边界值)和不合理的数据,进行测试,尽量使程序中的各语句和分支都被检查到,以便发现程序中的错误和漏洞,然后针对出现的问题,对算法和程序进行修改。

7. 文档编制

开发者编制程序文档的主要目的是为了让别人了解自己编写的算法和程序。文档包括源程序代码、算法(程序)的流程图、开发过程中各阶段的有关记录、算法(程序)的正确

性证明、程序的测试结果、对输入/输出的要求及格式的详细描述等等。

1.2.2 C 语言环境中的动态调试法

1. 单步执行和跟踪进入

使用“Run”菜单中的“Step over”命令或按 F8 功能键可进入“单步执行”方式,单步执行方法的特点是:程序一次执行一行,在执行完一行后就停下来,用户可以检查此时各有关变量和表达式的值,以便发现问题所在。但这种单步执行不能进入函数的内部,如果要进入到函数内部进行跟踪,则应利用“Run”菜单中的“Trace into”(跟踪进入)命令或按 F7 功能键。表 1-1 列出了 Turbo C 环境下 Run 菜单中各选项的功能。

表 1-1 Run 菜单中各选项的功能

选项	功 能
Run	编译,链接生成目标文件和可执行文件并运行。如果选择开关 Debug/Source debug 为 on,则程序执行到每一个断点都会暂停下来;再次选择该选项,程序将继续执行到下一个断点处或执行到结束处。
Program reset	用于动态调试。其作用是终止当前的调试操作,释放分配给程序的内存空间,关闭已打开的文件,但不改变断点设置。
Go to cursor	用于动态调试。使程序从执行行开始运行到编辑窗口中的光标所在行上。若光标所在行不含可执行代码语句,则显示一个 Esc 框,给出警告。
Trace into	用于动态调试。单步跟踪程序执行,并跟踪进入被调用的子函数,包括系统函数。
Step over	用于动态调试。单步跟踪程序执行,但不跟踪进入被调用的子函数。
User screen	显示程序的输出结果(把编辑窗口切换到程序输出窗口)。

2. 设置断点和监视表达式

在程序可能会出现问题或需要重点分析的位置,设置断点和监视表达式,程序执行到这里便暂停运行,这样即可通过观察变量值的变化对程序进行分析。

断点的设置方法:将光标移到某一行上,按 Ctrl + F8 组合键或使用“Break/watch”菜单中的“Toggle breakpoint”命令,此行就被颜色条覆盖,作为断点行。在一个程序中可以设置多个断点,运行时遇到断点行就会暂停,此时可以观察变量值的变化。如果想继续运行,再按一次 Ctrl + F9 组合键即可,程序会从暂停处继续向后运行。若要取消断点行,则将光标移到断点行上,再一次按 Ctrl + F8 组合键,此断点行即被取消;或使用“Break/watch”菜单中的“Clear all breakpoints”命令,这时程序中的所有断点行将全部被取消。

监视表达式的设置方法:按 Ctrl + F7 组合键或使用“Break/watch”菜单中的“Add watch”命令,会弹出一个添加表达式输入框。在输入框中输入表达式并确认,被监视的表达式即设置完成,当程序执行到这个表达式时,在屏幕下方的“Watch”窗口将显示表达式的值。若要删除一个或全部监视表达式,可以使用“Break/watch”菜单中的“Delete watch”命令或“Remove all watches”命令。

表 1-2 列出了 Turbo C 环境下 Break/watch 菜单各选项的功能。

表 1-2 Break/watch 菜单各选项的功能

选项	功能
Add watch	弹出一个 Add watch 窗口,在该窗口中输入监视表达式后,watch 窗口中将显示表达式的值。
Delete watch	删除监视窗口中最后加入的监视表达式。也可把光标移到 watch 窗口中的某一表达式处,用 Delete 键或 Ctrl + Y 键删除。
Edit watch	弹出一个编辑窗口,在该窗口中编辑最后加入的监视表达式。
Remove all watches	删除观察窗口 (watch 窗口) 中的所有表达式。
Toggle breakpoint	在光标所在行设置断点,再次执行该命令时删除该断点。
Clear all breakpoints	删除所有断点。
View next breakpoint	把光标移到下一个断点处。

1.2.3 常用的调试途径

1. 试探法

分析错误的症状,推测问题所在的位置,利用在程序中设置输出语句,分析内存中的内容等手段来获得错误的线索,一步步地试探和分析错误所在。这种方法效率很低,适合于结构比较简单的程序。

2. 回溯法

从发生错误症状的地方开始,人工沿程序的控制流往回追溯分析源程序代码,直到找出错误原因为止。这种方法对调试小程序较为有效,对于大规模的程序,由于其需要回溯的路径太多而变得不可操作,但可利用它单独调试大程序中的各个模块。

3. 原因排除法

包括对分查找法、归纳法和演绎法。

对分查找法:如果已经知道每个变量在程序中若干个关键点的正确值,则可以用赋值语句或输入语句在程序中关键点附近“注入”这些变量的正确值,然后运行程序并检查所得到的输出。如果输出结果是正确的,则错误原因在程序的前半部分;反之,错误原因在程序的后半部分。对错误原因所在的那部分再反复使用这种方法,直到把出错范围缩小到容易诊断的程度为止。

归纳法:首先把与错误有关的数据组织起来进行分析,以便发现可能的错误原因。然后导出对错误原因的一个或多个假设,并利用已有的数据来证明或排除这些假设;若已有的数据不能证明或排除这些假设,则需设计并执行一些新的测试用例,以获得更多的数据。

演绎法:首先设想所有可能的出错原因,然后试图用测试来排除每一个假设的原因。