

计 算 机 科 学 基 础

〔美〕 T. M. 沃克 著

周士观 刘国香 刘 棠 陆仁浩 译

内 容 简 介

本书是一本计算机科学的基础教科书。全书共分十一章。第一、二章主要讲授解题概貌及解题概念；第三、四、五章主要讲授解题和算法设计基础；第六、七、八章是把三至五章的概念应用于有关学科的题目中；第九、十、十一章主要介绍了各子算法的概念，并把子算法应用于求解物理学、数学和工程技术方面的问题，同时还介绍了计算机对经济、组织管理体制和对个人与社会的影响。

本书编写结构新颖，概念深入浅出，在各章末尾都有小结，在有关章节配有习题。本书可供高中、大学低年级学生、从事或想了解计算机的人以及其他工程技术人员阅读、参考。

Terry M. Walker
FUNDAMENTALS OF COMPUTER SCIENCE
Allyn and Bacon, INC.

计 算 机 科 学 基 础

〔美〕T. M. 沃克 著

周士观 刘国香 刘 荣 陆仁浩 译

责任编辑 陈永椿 曹美玉

科 学 出 版 社 出 版

北京朝阳门内大街187号

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1983年5月第 一 版 开本：787×1092 1/16
1983年5月第一次印刷 印张：17 1/2
印数：0001—44,000 字数：400,000

统一书号：15031·487

本社书号：3048·15—8

定 价：2.45元

译者的话

电子计算机的出现，对人类科学文化产生了巨大的影响。它的发展已经超出计算的范畴，渗入到各门科学领域以及日常生活中，成为现代化的一种标记。今天世界上不少先进的科学成就，如果没有电子计算机的帮助，是不可能获得的。所以自1946年第一台电子计算机在美国费城出现后，立即受到许多国家的重视。法国戴高乐亲自主持制定1966至1970年法国电子计算机的发展规划；日本从1966年开始研制超高性能计算机，由政府资助和大公司联合进行；西德制订了发展电子计算机的三个五年规划；英国也采取类似的研究部署。

美国是第一个制成电子计算机的国家，它不遗余力继续研究创新。三十多年来，电子计算机已发展进入到第四代，无论在功能、操作、体积大小等方面日新月异地不断向前发展，促进科学技术如乘风破浪般突飞猛进。近十年来，科学上的发展创新，电子计算机是起着主导作用的，大至对宇宙的探索，小至个人日常琐事的安排，工业、农业、交通运输等各个部门从质量到数量提高都发挥了空前的作用。

我国电子计算机的事业由于党的重视，也正在急起直追。为了迎头赶上，引进先进技术，培养人材，当是刻不容缓的事。译者有鉴于此，与几位中年教师和同志相互勉励，共同译出美国路易斯安大学教授T. M. 沃克所著《计算机科学基础》一书，以供读者参考。此书翻译系根据其1978年再版修订本，就这个版本来看，是按照目前普遍主张，将计算机科学的初等课程侧重于结合解题讲授，概念深入浅出，且着重于分析算法的思想逻辑，编写结构新颖，不但使学者知其然，且使知其所以然。士观年已九旬，脑力衰滞，学识又浅，出于为祖国实现社会主义现代化过程中尽我一点绵力，赖友人之共同努力译成此书，定有许多不妥之处，希读者予以批评指正为感。

译者 周士观

目 录

前言	vi
第一章 计算机的应用和解题	1
1.1 计算机的使用	1
1.2 解题基础	3
1.3 问题分析	3
1.4 算法设计	5
习题	10
小结	11
第二章 计算机系统引论	12
2.1 人作为信息处理机	12
2.2 计算机作为信息处理机	14
2.2.1 计算机硬件系统	14
2.2.2 计算机操作系统	21
小结	23
第三章 解题的要素和算法设计	25
3.1 问题分析	25
3.1.1 问题说明	25
3.1.2 问题的输入和变量	27
3.1.3 问题的输出	27
3.1.4 是要计算机解决的一个问题吗?	28
3.1.5 获得全部其它有关信息	28
3.2 算法设计	29
3.2.1 算法的性质	29
3.2.2 算法的表示法	30
3.2.3 算法已经有了吗?	33
3.2.4 算法可看作一个函数	34
3.3 在计算机上实现算法	36
3.4 算法设计中对计算机因素的考虑	38
3.5 程序文件	39
习题	41
小结	42
第四章 算法设计 I: 基本概念	44
4.1 流程图语言	44
4.2 常量、变量及说明信息	46
4.2.1 常量	46
4.2.2 变量和说明信息	47
4.3 终端操作、赋值语句及算术表达式	49

习题	53
4.4 输入/输出操作	59
4.5 算法流程及分支操作	63
习题	67
4.6 设计算法	68
4.6.1 最大值问题	69
4.6.2 折旧问题	78
4.6.3 元音计数问题	81
习题	84
小结	86
第五章 算法设计Ⅱ：循环、迭代算法和算法的效率	88
5.1 循环	88
5.2 下标变量	90
习题	95
5.3 三种精确的迭代算法	96
5.3.1 排序问题	96
5.3.2 查表问题	100
5.3.3 素数问题	103
习题	108
5.4 产生近似值的算法	109
5.4.1 平方根问题	110
5.4.2 近似值的收敛	116
习题	117
5.5 算法的效率	118
习题	120
小结	120
第六章 商业和公共事业管理方面的解题	122
6.1 商业数据处理	122
6.2 摊还表问题	125
习题	128
6.3 再论排序问题	129
习题	131
6.4 信用卡结算问题	132
习题	139
6.5 集中管理信息系统	140
6.5.1 集中信息系统一例	140
6.5.2 综合系统概念	142
小结	143
第七章 社会科学和艺术方面的解题	144
7.1 本文的统计分析问题	144
习题	147
7.2 使用统计学来概括数据	148
7.2.1 频率分布问题	149
7.2.2 概括统计量问题	153
习题	155

7.3 二元关系和预测问题	156
习题	159
7.4 社会科学用的灌装程序	160
小结	161
第八章 教育方面的解题	162
8.1 学做加法问题	162
8.2 专业选择问题	168
习题	176
小结	177
第九章 算法设计Ⅱ:子算法	179
9.1 子算法的基本概念	179
9.2 一般子算法	184
9.3 函数子算法	188
习题	192
小结	194
第十章 在物理科学、数学和工程技术方面的解题	196
10.1 整数函数和舍入误差	196
10.1.1 整数函数	196
10.1.2 舍入误差	197
习题	200
10.2 数字绘图问题	201
习题	205
10.3 方程求根问题	205
习题	211
10.4 求面积问题	213
习题	218
10.5 模拟问题	219
习题	224
小结	225
第十一章 计算机与社会	227
11.1 对经济和组织管理体制方面的影响	227
11.2 对个人和社会的影响	232
11.3 结论	235
习题	236
小结	237
附录A 纯数系统概念	238
A.1 纯数的记数	238
A.2 纯数的换算	240
A.3 特殊情况	244
A.4 补码	245
A.5 纯数的算术运算	246
习题	248
附录B 流程图数值函数运算符	251
附录C 参考书目	252
附录D 计算机术语主要词汇	254

第一章 计算机的应用和解题

第一台电子计算机是 1946 年制成的。自那时以来，计算机已大量应用于商业、行政机构、科学实验室和大学。计算机的广泛应用，使许许多多的职业应运而生。这种广泛的就业机会，使得千千万万的人都要求在计算机的设计和使用方面受到训练和教育。这种需要的结果，就产生了叫做计算机科学的这个学习领域。本书就打算介绍一些这方面的基本概念。

在这一章里，我们将首先了解一下计算机的一些用途。然后对涉及解题的一般概念进行探讨。这些概念是与作为本书主要研究对象的问题分析和算法设计概念有关的。

1.1 计算机的使用

商业上的应用

今天，计算机已用于社会的一切领域，执行多种多样的任务。实业界用它解决多种问题。例如，计算薪金帐和打印薪金支票。计算机在业务经营的使用上，差别最大的是生产厂矿所用的控制机。例如，用计算机控制轧机将钢坯轧制成钢板。再如，将计算机应用于班机航运服务系统（见图 1.1）。在这种情况下，计算机要存储关于飞行和乘航旅客的一切信息。

社会科学上的应用

目前，计算机已广泛应用于社会科学。在社会学领域应用的一例，是统计分析关于民族的大量数据，以验证涉及各社会集团的趋向和血统关系问题。心理学家可以在实验期间用计算机监测实验动物的动态。教育工作者可用计算机给学生讲授数学法则。计算机也已用来分析总乐谱。现在甚至已经致力于把计算机用于作曲方面。

物理科学方面的应用

计算机也被广泛地用于物理学、数学和工程技术方面。它已相当普遍地用来解各种方程和做诸如积分之类的数学运算。甚至物理学上对原子的粒子结构和动态的许多研究，也需要计算机来分析。纵览计算机应用的范围，实际上都包含着现代人的多方努力。

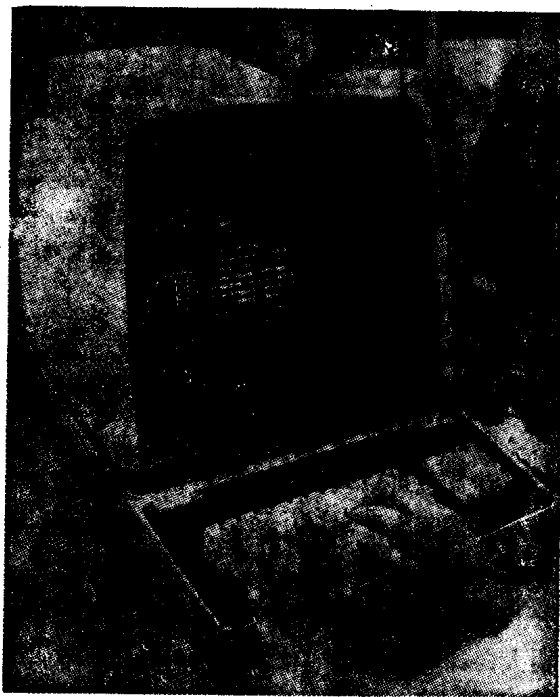


图 1.1 班机航运服务终端

计算机的能力

计算机应用日益广泛的一个重要原因是，它能帮助我们解决用其它方法不能解决的问题。载人的宇宙飞行和行星际宇宙飞行，已都能用上计算机，例如，只有通过高速地计算空间飞行器的运动，才能使必要的几乎瞬息之间的方向校正成为有效（见图 1.2）。计算机也用来处理银行支票。处理日益增加的支票这项任务（1974 年有 280 亿张支票是由联邦储备系统处理的）以及更新帐目，从经济上来说，没有计算机就无法进行。所以，如果没有计算机的话，银行就只好限制支票帐户的数目了。第三个例子是，计算机广泛应用于解决工程技术中和物理学中的矩阵问题。因为许多计算都要求去解决矩阵问题，所以，如果没有计算机，矩阵就只能侧重于理论方面的研究了。同样，在各种领域里，许多目前尚未解决的问题，借助于计算机也终将获得解决。

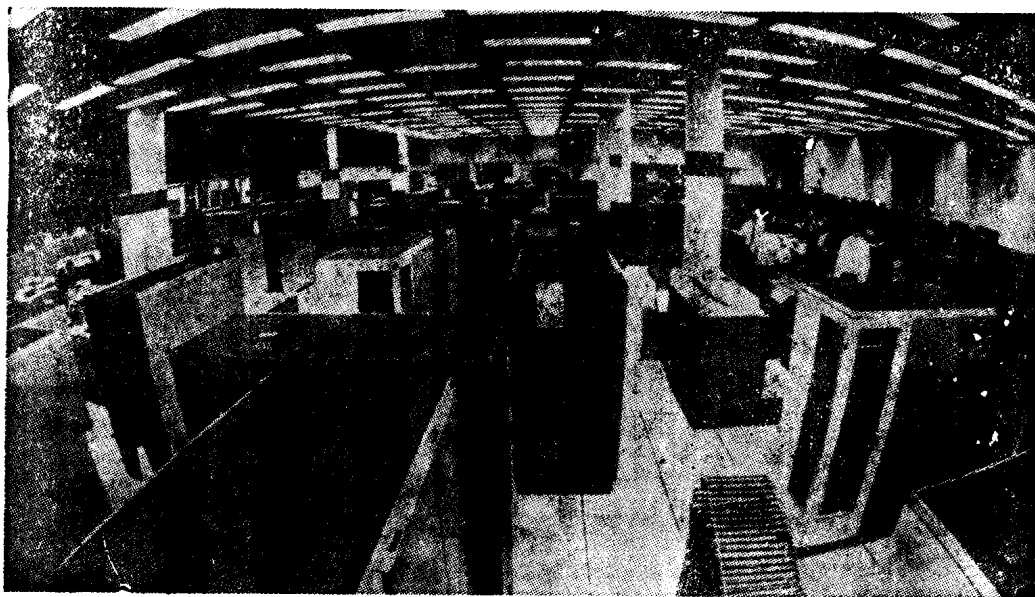


图1.2 得克萨斯州休斯顿国家航天局飞行器控制设备的各计算机

计算机的速度和精确度

计算机广为使用的另一个主要原因是，它解决许多问题能够比人做的快得多，也精确得多。大型计算机在进行计算时，每秒钟能完成 100 万条指令。而且它计算的精确度也高。因为有内部自行校验的特点，计算机不会在计算上出错，造成漏检。而人却做不到这点。计算机可以在几分之一秒的时间内累加数十万个各种大小的数而不出任何计算上的差错。作个不太恰当的比喻，这就好比：在一个月內，一个人在台式计算器上累加了十万个数字，却连一个错误也没出现！

但也应注意到，虽然计算机不会算错，可是它也可能出现不正确的结果。原因是，要由人向计算机提供它依次解决一个问题的步骤和它处理的数据。因此，无论解题方法或数据（或两者一起）一旦是错的，那就会出现不正确的结果。

计算机没有创造力

计算机即使有这么巨大的能力，可是它不能进行创造性的工作。计算机能够帮助建筑师设计一个建筑物，但是，没有建筑师，它却不能产生这项建筑设计。既然计算机承担了

它所能执行的任务，人就不必去做计算机所能做的事情了。如果我们命令计算机去做一些需要大量计算和数据处理的工作，我们就可以腾出手来处理新问题，并研究新问题以及现存问题的解答。

1.2 解题基础

我们在生活中，每天都会面对许多问题需要解决。我们着手解决其中多数问题的方法是很随便的。实际上，当我们在解决一个问题时，很少会想到我们这时已进入了解题活动。那是因为，我们这样做，往往只是下意识地解决其中的大多数问题。

问题的基本分类

我们每天都有问题需要解决，必需解决的问题有两大类，即：

1. 只出现一次的问题；
2. 出现一次以上的问题。

例如，对许多人来说，选择和谁结婚的问题，在一生中只发生一次。另一方面，我们许多人中，对于入学、离校，就业、离职等问题的选择却不得不进行上千次。有的似乎不常发生的问题，却往往是每一出现就又要得重新解决一次。那是因为，在解决这个问题时几乎没有想到在以前已经解决过这个问题。而经常出现的问题，许多人却只用一种解决方法（或有时用几种方法）。他们是在问题每次出现时都用这种方法去解决。例如，当我们每次决定要去刷牙的时候，可能都是经过同样的步骤。

我们常用这种粗略规定的方法解决每天的问题，这种非正式的过程已很令人满足。可是，当我们用计算机来解决问题时，就必须仔细地规定一套步骤，以导致问题的解决。这是因为计算机只能解决已经为之详细规定了一套解题规则的问题。制订这样一套步骤，当然必须随后有一个比较正式的解题过程来完成它。

在这章的后一部分，我们将继续讲一些一般的解题概念。这将使我们开始考虑一个关于形成解题方法过程的比较正式的途径。在第二章里，我们就要着手对于计算机的组成情况以及用以解题的方法做个深入的探讨。然后，我们就准备更深入一步研究制定计算机解题方法的过程。

1.3 问题分析

问题说明

对任何解题过程的一个很重要的方面，是对问题的分析。问题分析是对问题加以说明的过程，是找出为解题所需的一切信息的过程。对一个问题予以恰当说明的重要性怎样强调也不为过分，因为对错误的问题制定一个解决方法，是没有意义的。解决问题的方法无论多么好，而所解决的却是个不对头的问题，这种事是有的。比如，问题是要做巧克力饼，那么，即使最好的苹果馅饼烹制法也不能解决我们的问题。如果我们的问题实际上也许只是要准备一个甜点心。那么，上等苹果馅饼烹制法恰好就能解决这个问题。论点很简单，一个不对头的问题的最好的解决方法，对于有某个问题需要解决的人来说，是毫无价值的。

适合于解决所提出的问题的方法，往往有无数个。可是，为了选择一个最好的方法，我们必须首先对要解决的问题给予恰当的说明。

让我们另举一个例子，再进一步讲一讲问题，说明这个论点。假设我们的问题是选择一条驱车去当地馅饼店的最好的路线。为建立一个解决这个问题方法，我们可能要花很多的时间。但是，这可能不是我们真正想要解决的问题。问题实际上可能是找一个去当地馅饼店的最好的走法（注意，不是路线，而是走法）。在这种情况下，就有好几种解决方法可以选用。这里面包括乘公共汽车或乘出租汽车，或请一位朋友用车带上我们去。

说明的精确性

由于问题说明的表达必须精确，因而说明一个要解决的问题的过程就更形复杂。例如，在选择一条驱车去当地馅饼店的最好路线的问题上，“最好的”这个词是含糊的。说“最好的”这个词含糊，理由是因为我们不知道从何种意义上说一条路线是最好的路线。最好的路线可能是行车距离最短的路线。或者可能是行车时间最短的路线。第三种可能是，这条路是遇到交通信号最少的路线。显然，从其它意义上的理解，还可以提出许多最好路线。此外，最好的也可能包括上述的几个判断标准。所以，问题说明的一个重要部分，就是用精确的术语来说明问题。这一点是很必要的，这样就可以使我们避免由于误解需要解决的问题而建立解决错误问题的方法。

子问题

精确地说明问题以便为之找到解决的方法，这只是问题分析的第一步。问题分析还包括研究可能涉及制定解决方法的一切子问题。例如，在去馅饼店最好路线的问题中，在某些点，我们须对下一步选择的路线做出判断。比如，交通阻塞可能使我们走另外一条通往馅饼店的道路。为此，采用的几条路线和我们采用的其中每一路线的情况都必须摆出来。再举另外一例，在计算工资表的问题上，只知道应从工资中扣除社会安全税是不够的。我们还需注意到社会安全税只从每年至今为止的所得中扣除，这比已定的总数为少。

输入和输出

问题分析的另一个方面是验证：

1. 一切输入都是解题所必需的；
2. 一切输出都要求是问题的解。例如，我们来检验一下烤巧克力饼的问题。输入是制做这种饼的用料，包括面粉，油和巧克力香料。输出是巧克力饼。再如，去馅饼店的最好路线问题，输入是时间，将遇到的交通拥挤情况以及气候条件等这类的情况。这个问题的输出则是到达馅饼店。

至此我们也许才开始看到，问题分析的整个领域实际上都是很主观的。总是能提供解决该问题所需一切信息的分析问题的什么好方法，是不存在的。我们扼要地讲一下问题分析的过程，简单地说，它包括：

1. 制定问题的精确说明；
2. 找出一切子问题；
3. 列举一切必要的和有效的输入；
4. 列举一切想要得到的输出。

在第三章中，我们将更深入地探讨与问题分析有关的一些概念。

1.4 算法设计

算法

在一个问题已经完全分析完毕后，我们就准备制定一种方法来解决它。这个解决方法的具体化，我们就称之为算法。算法是由一套明确的规则组成的一些步骤，它指定操作的顺序，将问题按一定数目的步骤加以解决。下面我们将看到，算法可以用好几种方法来表示。简单地说，算法是能使问题得到解决的一个逻辑顺序。

虽然上述对算法的定义可能听起来很复杂，但我们却经常使用算法。巧克力饼的烹制法就是算法的一例。象这样一个烹制法，只包括一套规则：

1. 按一定顺序将定量的各种用料和在一起；
2. 把和好的用料放在炉灶上烘烤一定时间。

组成这种烹制法的规则必须明确。同样，这些规则必须在经过了一定步骤之后，最终输出巧克力饼。巧克力饼烹制法大都可以满足这些要求。

算法的第二个例子是一个模型飞机装配指令表。这个指令表告诉我们，按一定顺序把哪些部件装配在一起，最后装成一个飞机模型。再者，这些步骤必须切实固定，而且必须明确（不含糊）。执行这样的指令表的结果就该是一个模型飞机。

去托尼路线的第一个算法

就算法再举一例：编排一下我们驱车去最受欣赏的馅饼店所循路线的顺序。假定我们愿去琼斯顿大街上托尼馅饼店吃馅饼，那么，其算法可列出如下：

1. 离开我们的宿舍，走向停车场上我们的汽车；
2. 上车并扣紧座椅的皮带；
3. 开动汽车的发动机；
4. 开出停车场驶向宿舍前的大街；
5. 向高速公路驶去；
6. 沿高速公路向前行驶五英里；
7. 在邮筒处驶离公路并向右转；
8. 驶往馅饼店停车场。

我们是否乐于接受这个算法所表示的路线呢？对这个问题的回答必定是不，因为上述过程不是由明确的规则组成的。比如，第5步没有谈到一旦我们驶入大街，高速公路在哪个方向。其次，我们行驶的是哪一条高速公路也不明确。在我们宿舍附近也许有两、三条公路。第6步至第8步也有含糊之处有待更改。

我们也许会问，如果我们懂得每一步是怎么回事，这种含糊不清之处有什么要紧呢。对这个问题的回答是，我们常常要使一个算法能让任何希望使用这个算法的人都能用来解决他的问题。比如，在我们楼下住着的一位朋友，可能也要到琼斯顿大街托尼馅饼店去吃馅饼。此外，他可能想用我们的算法驱车去吃馅饼。因为他必须使用这个算法，那么，我们的算法就一定要精确到足以使他达到他的目的。那就是说，我们的朋友靠使用我们的算法定能到达托尼馅饼店停车场。

在后面的几章里，这种算法步骤含糊的不足之处，就显得重要了。这是由于我们设计出来的所有算法，都将在计算机上实现。在第二章里我们就会知道，计算机必须有完全精确的步骤可遵循，才能解决问题。

去托尼路线的第二个算法

为解决去托尼馅饼店路线的问题，我们第二次又试做了一个算法，我们可以：

1. 离开我们的宿舍，走向停车场上我们的汽车；
2. 上车并扣紧座椅的皮带；
3. 开动汽车的发动机；
4. 开出宿舍停车场，驶向埃姆大街；
5. 沿埃姆大街北行，经四个街区到1-10高速公路入口处；
6. 在通往高速公路上跨通道前右转弯，驶上高速公路入口的斜坡；
7. 沿1-10高速公路东行，到琼斯顿大街出口处（这里是第八个出口处）；
8. 右转弯驶上琼斯顿大街；
9. 沿琼斯顿大街南行经六个街区；
10. 从琼斯顿大街向右转弯驶入托尼馅饼店停车场，把车停在适当的空位上。

对这个算法，我们首先注意到的是，这一套规则相当明确。这些规则对于我们楼下的那位朋友，就象对于我们一样，是有意义的。在这些算法步骤中，我们已提出了明确的街道名称，公路号码，还有方向。这与前面提到的我们去托尼馅饼店路线的算法中那些很一般的、含糊的规则相比，就大不相同了。

上述去馅饼店循行路线的算法是用自然语言表示的。用这种方法来表示一个算法，可能只适用于解决象选择去托尼馅饼店路线这样比较简单的问题。可是，在算法比较复杂的情况下，自然语言终究是个不令人满意的表达手段。不宜用自然语言表示算法，可以提出好几个理由。首先，自然语言含糊不确。比如，象 love（爱）这个词，会使人联想到许多意思。其次，用自然语言表示一个算法，使算法的逻辑流程不易展开。本章稍后，当我们设计包括一些分支的算法时，这一点就会显得更为重要。

流程图语言

在计算机科学中广泛用来表示算法的一种语言，就是流程图语言。本书所使用的流程图语言包括：

1. 一套八个标准流程图符号，其中六个，我们将称之为流程框；
2. 英语字母表的26个大写字体和26个小写字体符号，有关的标点符号，还有一些特殊符号；
3. 十进制的十个数字符号，有关的算术运算符号；
4. 把这些符号汇编成算法语句的规则。每个流程框的形状都表示一种操作类型，操作就在框内执行。关于其它符号和规则及将其汇编成算法语句的问题，将在本章稍后，以及在第四、第五和第九章中详细说明。

去托尼路线算法的流程图

我们第二次做出的去托尼馅饼店路线的算法，用流程图语言的表示法可见图1.3.。在这个流程图中，只用了四种流程图符号：

1. 椭圆框;
2. 矩形框;
3. 圆符号;
4. 带指示箭头的直线。

请注意, 流程图的每个框在其右上方都有一个小的编号。这些都是简单的表示顺序的数, 它使我们便于参阅, 比如, 流程图中的第三框, 就是框3。

终端框

让我们在这里穿插着详细地讨论一下这个流程图。首先, 你们会注意到框1和框12都是椭圆的。一般来说, 椭圆框表示算法中的终端。也就是说, 算法中的起点或终点都同样用它。所以, 框1表示算法中所要履行的第一步。同样, 框12则表示问题至此已得到解决, 算法已到终点。我们就把椭圆形的框叫做终端框。

处理框

框2至框11都是矩形的。它们表示在执行算法期间所完成的各种处理操作。这种矩形框叫做处理框。

连接符

带有数字的两个圆符号叫做连接符。连接符的含意是表示流程的延续。连接符有两种:

1. 引出连接符;
2. 引入连接符。

引出连接符是一个带一条线的圆圈, 并有一个指向连接符圆圈的箭头。引入连接符也是一个带一条线的圆圈, 但箭头指向一个流程图的框框或指向一条流线。引出连接符表示流程重新回到流程图的某一部位; 引入连接符是流程图中流程恢复的点。比如, 在图1.3中, 左边纵排符号的底部有个内有数1的连接符, 就是引出连接符。它表示流程重新恢复从某个内有数1的引入连接符开始。而引入连接符则表示流程图中已由某处开始的流程顺序的延续。在这个例子中, 延续点就是流程图中在右边纵排符号顶部的引入连接符。

流线

最后, 这个流程图中的符号都是互相由带有指示箭头的直线连接起来的。这些指引方向的直线就叫做流线。流线用来表示执行算法步骤所依循的顺序。

去托尼馅饼店路线算法的第二种流程图

图1.4是我们去托尼馅饼店路线算法的第二种流程图。也是图1.3中所示算法的另一种说明图。这个图提出的新东西是判定过程, 这在图1.3的流程图中是隐示的。在图1.3的框6中, 我们指示执行算法的人沿埃姆大街北行至1-10高速公路入口处。在这个指令中

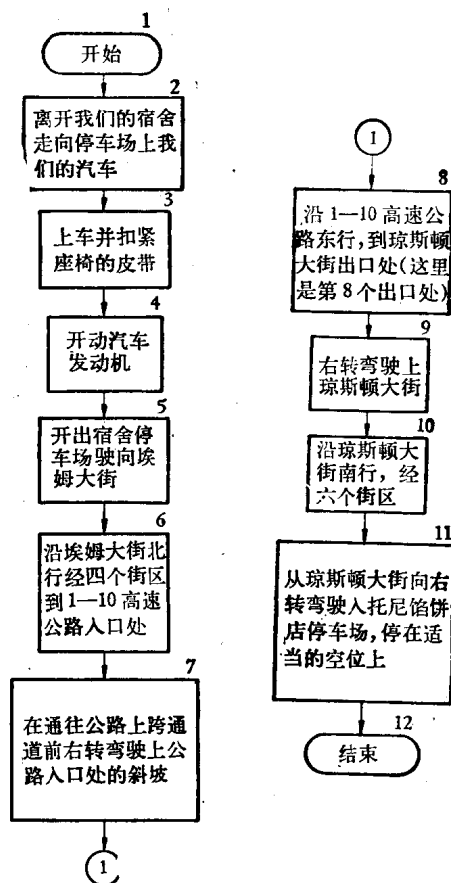


图1.3 去托尼馅饼店路线流程图

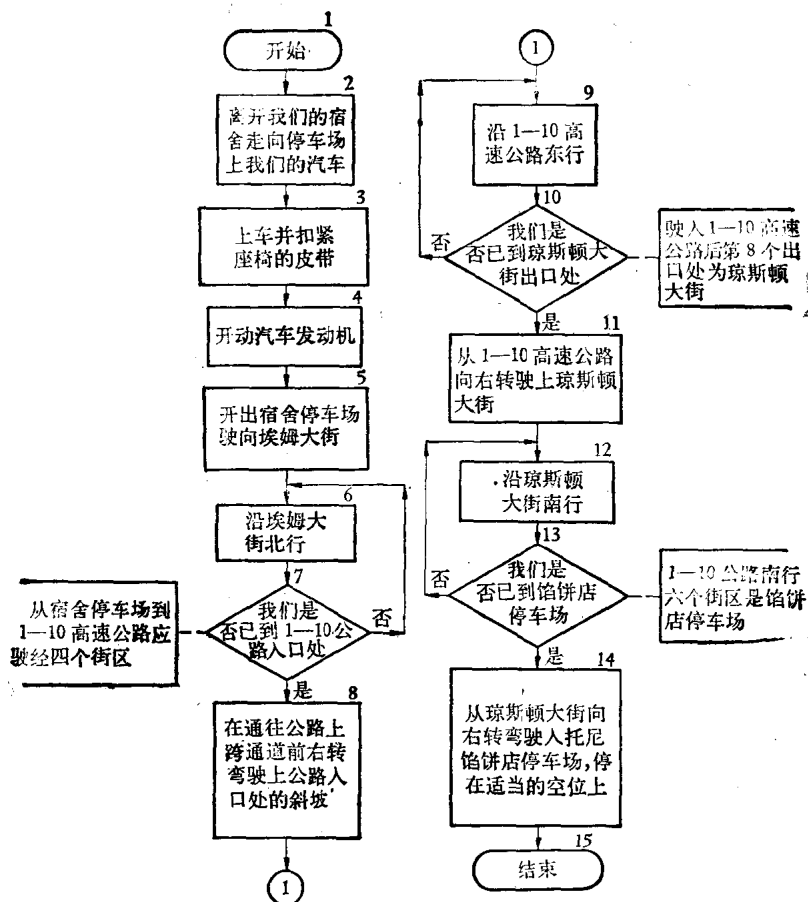


图1.4 去托尼馅饼店路线的第二种流程图

暗示：这个人仍然需要注意是否已经到了入口处。如果我们还没有到入口处，就继续沿埃姆大街北驶。在到达入口处时，我们就向右转弯开上跨通道引道前的高速公路入口斜坡。

判定框

为此，原流程图中的框6在图1.4中扩展为框6和框7。这一步的扩展是要明确表示在图1.3中是隐示地做出判断的过程。新流程图的框6包含的是一个行为，而框7包含的是一个判定问题。图1.3中的符号8和10同样也做了这样的扩展。可以看到，这种判定框是一个菱形的符号。

循环

重复执行算法步骤的过程（如图1.4流程图中框6的重复执行）叫做循环。在循环中，有一个或多个算法步骤进行重复，直到某个判定标准已达到时为止。关于循环这个论题，在第五章中还要做比较深入的探讨。

注释框

最后，可以看到，图1.4中出现的流程框的另一新形状。这种框是一个一端开口的矩形符号。这种框叫做注释框，是用来记述或解释信息的。我们注意到在这些框上没有框号。另外，注释框不用流线，而都是用虚线与流程图中的其它框相连接的。这是因为，注

释框不属于应执行的算法步骤。所以，它们不在算法的正式流程之内。注释框只是对有关的各个算法步骤赋予解释信息的手段。图 1.4 中的几个注释框分别与框 7、框 10 和框 13 相连接。在每个这种框里都得到一个信息，说明到每个算法循环所形成的出口点的地理上的距离。

测验评分示例

最后我们举个例子来研究一个算法：取五次测验评分，把五次评分加在一起，把和记录下来。这个算法的流程图如图 1.5 所示。我们可以看到，在框 2，3，4，5，6 和框 14 中，出现了一种新形式的框，即平行四边形。

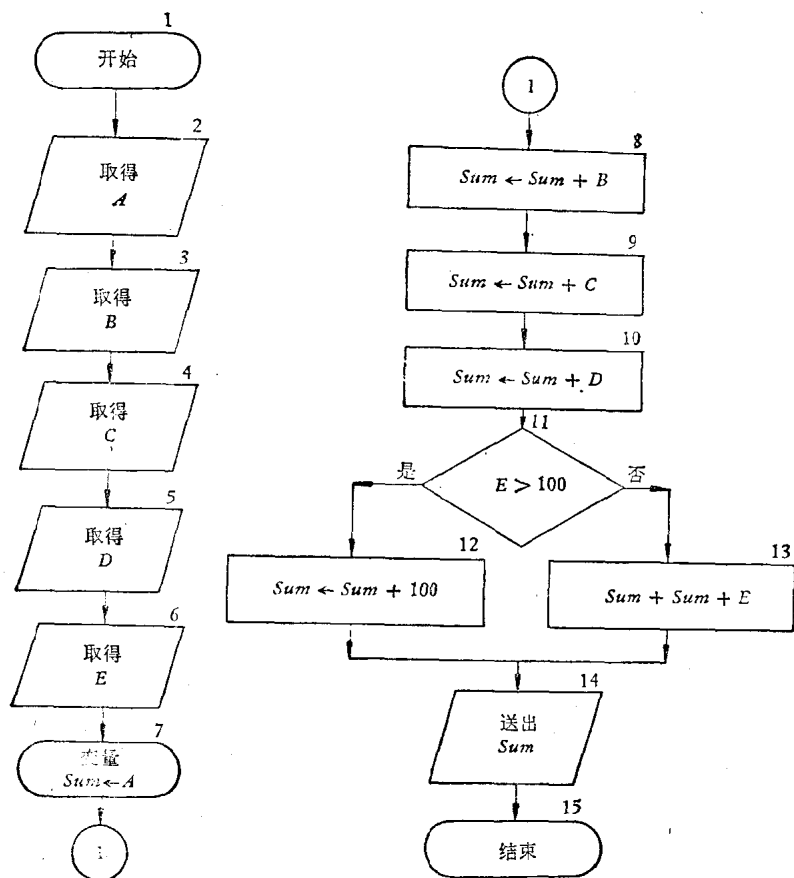


图1.5 五次测验评分算法流程图

输入/输出框

这种平行四边形的框叫做输入/输出框，比如表示把数记录在纸上，以便继续操作这样的操作方法。我们在使用输入/输出框时结合使用两个动词：“取得”（GET）和“送出”（PUT）。

当一个输入/输出框中出现动词“取得”（GET）时，就是说对算法有一个输入。例如，图 1.5 中，某人给我们读五个数值输入给算法。在流程图的框 2 至框 6 中，我们展示了把这些数记录下来的操作，并且把这些数用变量 A, B, C, D, E 表示。一经录下，

* 变量和常量将在第四章中作较充分的探讨。

我们就能把用 A, B, C, D, E 为变量名称的符号与这些数值相对照。

在输入/输出框中出现动词“送出”(PUT), 就是说算法中使用的数值已被记录在某个媒体上。拿我们这个算法为例, 就是把变量“Sum”的数值写在一张纸上, 以便交给指令执行者。这项操作是在图 1.5 的框14中完成的。

赋值操作符

得出“Sum”所必需的步骤见框 7 至框10以及框12和框13。我们看到其中每一个框内都有变量名称“Sum”, 它的后面有一个左指箭号。这个左指箭号就叫做赋值操作符, 因为它被用来赋予变量以数值。在这个例子中, 数值被赋予名为“Sum”的变量。在每个框内赋值操作符的右面或是一变量, 或是一简单的算术表达式。算术表达式往往是表示几个变量中一个变量的值加上称为“Sum”的变量的值。

在框11中, 必须判定第 5 次测验评分(变量 E 的值)是否多于 100。因为第五次测验包括一个附加分数问题, 它使分数的总数可能多于100。不过, 对于第五次测验, 教师不想使分数多于100分。为此, 如果变量 E 的值大于100, 就执行框12, 使得只有 100 分被加进“Sum”。否则, 当第五次测验评分已确定是不多于 100 (小于 100 或等于 100), 就执行框 13。这使得 E 的值被加进和数。应注意, 或是执行框12, 或是执行框13; 决不会是两者都执行或两者都不执行。

本节和前各节的讨论使我们获得了一些关于解题和算法设计的基本概念。但是, 我们还能制定一个计算机能用来解题的算法。这要在第三章开始做比较详细的讨论。

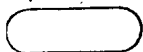
习 题

1. 图1.4的算法另外还有什么含糊不清的地方吗? 如有, 就将算法做必要的改动。
2. 制定一个寄信算法的英语表达式。这一算法就从你刚在信末签名之时开始。假设: 这封信要投入距离两个街区远的邮政信箱。
3. 用本章的流程图符号, 为习题 2 的算法绘制成流程图的表示形式。
4. 制成一个乘汽车途中换轮胎算法的英语表达式。这一算法的开始点是, 你刚驶离公路并发现有一个轮胎漏气了。算法中可以考虑到一切意外情况, 包括备用轮胎也已漏了气的可能性。
5. 为习题 4 的算法制成一个流程图表达式。
6. 为给门上一个被打破了的玻璃窗换玻璃, 制定一个算法的英语表达式。假设: 玻璃块还留在原处, 四周有小木条, 每根木条上钉有五个小钉子。
7. 为习题 6 的算法制成一个流程图。
8. 为打一个长途电话制成一个算法的英语表达式。考虑到涉及安排一次长途电话的一切通常采用的办法。
假设: 你用的是按键电话, 而且已经知道你想要的电话号码。
9. 为习题 8 的算法制成一个流程图。
10. 为穿衣服制定一个算法流程图。包括你要到什么地方去的问题, 作为本题的一部分。
11. 为星期六晚上赴约制定一个算法流程图。
12. 按下述情况修改图 1.5 的算法流程图: 在第一次和第三次测验时也含有附加分数问题。假设, 在任何一次测验时分数都不多于100。

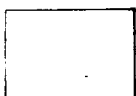
小 结

1. 人们通常用来解题的方法都是无意识地完成的。
2. 问题有两大类:
 - a. 只发生一次的问题;
 - b. 发生一次以上的问题。
3. 要解决一个问题, 计算机要求有确切说明的一套步骤。
4. 为了使用计算机解题, 要有正式的解题概念。
5. 问题分析是一个过程, 通过这个过程, 问题得到说明, 而且解题所必需的一切信息得到认定。问题分析包括:
 - a. 制成一个问题的精确说明;
 - b. 找出一切子问题;
 - c. 列出一切必须的且可以采用的输入;
 - d. 列出一切希望得到的输出。
6. 对错误的问题, 最好的解也是无用的。
7. 一个算法是一个过程。这个过程是由一套毫不含糊的规则组成的, 其中详列按有限数目的步骤解题的操作顺序。
8. 一个算法应当使任何一个希望使用这个算法的人都能解决问题。
9. 算法往往用流程图语言来表达。
10. 本章所介绍的流程图符号为:

a. 终端框:



b. 处理框:



c. 连接符:



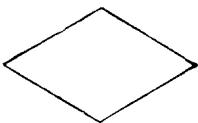
d. 流线:



e. 输入/输出框:



f. 判定框:



g. 注释框:

