算法设计与分析

原福永　等编著

机械工业出版社

算法设计与分析

原福永　张玉连　刘玉峰　张英慧　编著
刘国华　主审

机械工业出版社

算法研究是计算机科学的核心课题之一。《算法设计与分析》是计算机硬件、软件专业同时包括应用专业的基础课程，对从事计算机系统结构、系统软件和应用软件设计是非常重要和必不可少的。本书共计十章，围绕算法设计的基本方法和算法分析的基本理论，对计算机领域中常见的算法作了简捷、通俗的描述和系统、深入的分析。本书特点有二，一是通过对计算机领域中许多常见而有代表性的算法的研究，使读者理解和掌握算法设计的主要方法，培养对算法复杂性分析的能力；二是突出面对一个问题时，有一个有效的解题思路。

本书以计算机专业师生为主要对象，也可供自学者和科技人员参考。

图书在版编目（CIP）数据

算法设计与分析/原福永等编著．-北京：机械工业出版社，1998.6
ISBN 7-111-06581-6

Ⅰ.算… Ⅱ.原… Ⅲ.①电子计算机－算法设计②电子计算机算法分析 Ⅳ.TP301.6

中国版本图书馆 CIP 数据核字（98）第 16363 号

出 版 人：马九荣（北京市百万庄南街 1 号 邮政编码 100037）
责任编辑：杨明强 版式设计：杨丽华 责任校对：李 悦
封面设计：李 明 责任印制：侯新民
北京市昌平精工印刷厂印刷·新华书店北京发行所发行
1998 年 7 月第 1 版·1998 年 7 月第 1 次印刷
787mm×1092mm 1/32·9 1/4 印张·202 千字
0 001—1 000 册
定价：17.60 元

前　言

算法研究是计算机科学的核心课题之一。算法研究的目的是要设计求解速度快、占用空间少和设计时间短的解决问题的方法。《算法设计与分析》是计算机硬件、软件专业同时包括应用专业的基础课程，对从事计算机系统结构、系统软件和应用软件设计是非常重要和必不可少的。

本书是以计算机专业学生为主要对象编写的。通过非数值的数据对象和串行方式，介绍了如何设计和分析计算机科学中常用的基本算法。所涉及的相关知识主要包括离散数学、程序设计和数据结构等内容。

本书的目的有两个，一个是通过对计算机领域中许多常见而有代表性的算法的研究，使读者理解和掌握算法设计的主要方法，培养对算法复杂性进行分析的能力，为独立设计算法和分析算法奠定基础。另外，培养读者在面对新算法时能够有一个良好的思考习惯，这个算法效率怎样，是否有更好的算法？在面对一个问题时有一个有效的解题思路。正确分析该问题，考虑能有几种途径解决该问题，并且比较它们的好坏，在修改和放弃之间做出选择，直至得到满意的结果。

本书共分十章。主要内容有算法的基本概念和基础知识，常规的算法设计方法和算法分析技术，最后介绍了 NP 理论。

本书由燕山大学刘国华副教授主审。

本书在编写过程中得到了燕山大学赵树春教授、王德滋

副教授和吴卫江研究生的热情帮助，在此一并表示衷心感谢。

由于编者水平有限，缺点和错误在所难免，望读者指正。

编　者

1998 年 5 月

目　录

第一章 绪 论

在本章中，首先介绍算法的基本概念和性质，其次给出评定算法常用的标准，以及算法如何描述，如何分析，最后对算法分析所要做的具体工作进行了一般的介绍。

§1-1 引言

在日常生活中，我们会遇到许多问题，对于这些问题，我们要采取一定的方法解决它。就某一问题而言，解决方法的具体化，就可以看作是解决这一问题的算法。有程序设计经历的人对“算法”一词是熟悉的。因为程序设计的第一步就是确定算法，它被理解为需要解决的问题与编制程序间的一个步骤，在这里算法代表着用系统的方法描述解决问题的策略机制。下面我们给算法概念一个非形式的描述：一个算法是一个有穷规则的序列。这些规则确定了解决某一类问题的一系列运算。对于这样一个序列，如果有一组输入，它能进行有限步计算并产生一组输出，然后终止。

一般地讲，算法必须具备以下五个重要特性：

1. 有穷性

一个算法在执行有穷个计算步后，必须终止。不能终止的过程不属于算法的范畴。

2. 确定性

一个算法中给出的任何计算步必须有确切的含义。即计算步骤要执行的动作是清楚的、无二义性的。算法中不允许

存在可被解释成不同含义的计算步。

3. 能行性

一个算法中要执行的任何计算步都是可被分解为基本的可执行的操作步，即每一个计算步都可以在有限时间之内完成。

4. 输入

一个算法有零个或多个输入，它们通常被解释为算法运行的初始数据。它们取自特定的集合。

5. 输出

一个算法有一个或多个输出，它们是同输入有某种特定关系的量。它们通常被解释为输入的计算结果。

凡是一个算法，都要具有以上五个特性，缺一不可。否则，不能称为算法，而只能看成是过程、程序等。

例 1.1　已知两个正整数 m 和 n，求它们的最大公因子。

我们可以用辗转相除法（也叫欧几里德算法）求解。描述如下：

第一步　求余数　以 n 除 m，令 r 是所得的余数，$0 \leqslant r < n$。执行第二步。

第二步　判余数　如果 r=0，输出 n 的当前值，结束。否则，执行第三步。

第三步　改变被除数和除数　m←n，n←r，转去执行第一步。

这一方法用流程图的形式可描述如图 1-1。

我们如果用程序设计语言来描述这一方法，还可以给出下面的程序：

```
PROCEDURE EUCLID;
  VAR
```

```
  m, n, r: integer;
begin
  read (m, n);
  while n≠0 do
    begin
      r←m mod n;
      m←n; n←r;
    end;
  write (m)
end;
```

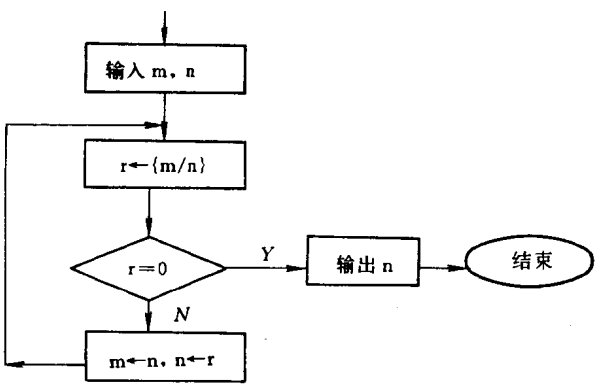

图 1-1

让我们来看一下上述计算过程是否可以称为算法。该过程规定了三个计算步骤，每个步骤意义明确、可行。第三步的循环终止条件是余数为零。因为余数总是小于除数的，所以余数等于零的情况在若干次循环后一定会出现，即具有算法的有穷性。这个计算过程也具有算法的输入和输出的特性，所以上述计算过程是一个算法。

例 1.2　在整数集合 s 中找出最大元素。

若 s 中的元素用数组 A [1: n] 表示，则该问题可用下面的过程求解。

```
PROCEDURE MAX;
  VAR
    i, m, n: integer;
    A: ARRAY [1..n] of integer;
  begin
    read (n, A [1], A [2], ..., A [n]);
    i←1;
    while  i<n  do
      begin
        if  A [i] >A [i+1]  then
          begin
            m←A [i];
            A [i] ←A [i+1];
            A [i+1] ←m
          end;
        i←i+1;
      end;
    write (A [n])
  end;
```

这个过程有算法应具有的五个特性，所以，它也是一个算法。

§1-2 算法的描述

算法实际上是一个抽象的概念，它的描述是基于语言的。一般情况下，算法可以用以下几种方式描述。

1. 自然语言

我们日常所用的语言就是自然语言，如汉语、英语等。在例 1.1 中第一种描述方式用的就是自然语言。使用自然语言的主要优点在于语言自身是我们熟悉的，但由于语言自身的原因，使得它表现出的缺点难以克服。一是语言自身有二义性带来对语句的理解产生二义；二是语句一般都较长，导致写出的算法太长，阅读困难；三是自然语言的表示法是自然串行的，对于分枝和循环等逻辑结构不能清晰地显示；再者，目前计算机还不能直接处理自然语言描写的算法。

2. 计算机程序设计语言

在例 2.2 中算法就是用计算机程序设计语言描述的。用计算机程序设计语言描述算法的优点是清晰、简明和能用计算机处理，不足之处在于首先要掌握用于描述算法的计算机程序设计语言，另外，算法逻辑在程序中不能完全遵循，这会带来描述上的困难和交流上的不便。

3. 流程图语言

在例 1.1 中，第二种描述方式就是流程图语言。用流程图语言描述算法的主要优点表现在算法的逻辑结构明显，算法易于建立和易于理解，另外，一个用流程图语言表示的算法能容易地翻译成计算机程序设计语言表示的算法，为算法在计算机上的执行和修改带来了极大方便。它是我们使用比较多的一种描述方式。今后我们将主要以计算机程序设计语言和流程图语言两种方式描述算法，以满足算法设计和算法

分析的要求。这样，我们今后经常看到的算法就会是一个计算机程序。

事实上，一个算法不等价于一个计算机程序或一个过程。因为，算法是一定要终止的，在这一点上算法是有穷的过程，而过程不但可以是有穷的，也可以是无穷的。我们说算法与计算机程序不等价，是因为一个计算机程序往往是指用某种计算机程序设计语言所写出的一个计算过程，但一个特定的算法不一定表现为一个计算机程序，二者没有必然联系。一个算法可以用多种方式描述，如用图解描述，语言描述等。如果用语言描述一个算法，既可以用人类的自然语言，也可以用各种计算机程序设计语言。只有在用计算机程序设计语言描述时算法才表现为计算机程序。再有，一个计算机程序只限于在计算机上运行，而一个算法的运行不受这个限制，它可以编成程序在计算机上运行，可以用纸和笔手工运行，也可以用其它工具运行。因此，我们说算法较之计算机程序有描述的多样性和多种实现方式。

§1-3 算法的设计与分析

如果一个问题是可解的，那么求解的全过程应由两个阶段构成，第一阶段是要设计出解决问题的算法，第二阶段是运行算法并获得解。在选择了描述语言之后，算法就可以用该语言适当地表示出来。当然，把算法描述成计算机程序，需要我们熟悉计算机程序设计语言。利用计算机实现算法是强有力的手段，我们不能不借助计算机这个工具。所以，我们较多地利用计算机程序设计语言来描述算法，并分析算法在计算机上执行的情况。

算法设计和分析的步骤可概括为：

1. 问题的陈述

为了设计求解某一问题的算法，首先要把已知什么和要求什么陈述清楚。问题的陈述虽不困难，但却是重要的。没有清晰的陈述是难于设计出好的算法的，有时甚至设计不出正确的算法。

2. 选择模型或拟制模型

当问题陈述清楚后，接下来要选择或拟制描述问题的数学模型。这项工作也是重要的，模型适当与否影响算法设计的速度和算法的效率，关系到后续工作的顺利与否。

3. 算法设计和确认

数学模型确定以后，就可进行设计算法的工作，它是一种复杂的工作，也是一种创造性工作，它是不可能完全自动化的，也不会有万能的算法设计方法适于解决一切算法设计的课题。掌握一些基本设计策略，对设计算法是十分有益的，也是十分必要的。一旦设计出了算法，就应当证明它对所有可能的合法输入都能算出正确的答案，这一工作称为算法确认。确认的目的在于使我们确信这一算法将能准确无误地工作。算法确认亦即算法正确性证明是一项艰巨繁杂的工作，它是要证明对一切合法输入，算法都能产生正确的输出。通过穷举法可以证明算法是否正确，但当合法输入是无穷时，穷举证明是不可能的。通过下面的方法也可以证明算法是否正确。将算法的输入和输出分别表示为“输入断言”和“输出断言”，计算步表示为一组“谓词演算”规则，正确算法的“输入断言”通过它的一组“谓词演算”规则能够导出“输出断言”。这个论证过程有时比算法设计过程还复杂。一般情况下，我们并不去严格地证明算法的正确性，而是采用并不严密的非形式的直观解释来代替这一证明。

4. 程序实现

程序实现就是将一个算法正确地编写成计算机程序。一旦证明了算法的正确性，就可将其写成程序。这项工作看似简单，但要求程序设计者有很好的程序设计基础，掌握多种程序设计方法和技巧，这样才能避免算法走样，使程序有较高的运行效率。转换后的程序是否正确也是需要证明的，这一工作称为程序证明，它应在程序运行前完成。

5. 算法分析

算法分析是算法设计完成后的一项必然性工作，它是研究各种算法的特性和好坏的。对于解决同一类问题或许可以设计出若干个算法，人们自然想知道这些算法哪一个好，哪一个差。如果存在优劣差别，如何才能判定它们的这种差别？有什么样的标准？这些都是算法分析要回答的问题。

对于算法要分析什么？怎样分析？在回答这些问题以前，我们先来看看评定算法优劣的标准是什么。

评定一个算法我们通常从四种观点出发，采用以下五条标准进行。

(1) 时空的观点

标准 1　算法在计算机上执行的时间最短。

算法执行时间的长短，主要决定于算法的逻辑，不同的设计思想所产生的算法差别会是很大的。其次，程序设计的水平也会反映到算法执行时间的长短上。不必要的输入输出操作，公共处理部分安排的不合理和循环控制的不准确都会使执行时间增长。

标准 2　算法需要的存储空间最小。

这主要考虑安排合适的数据结构。一般情况下，这会带来执行时间的增加。标准 1 和标准 2 有时是矛盾的。

(2) 发展的观点

标准 3　算法的适应性最强。

如果一个算法对目前和将来的适应性强，说明这个算法的效率是高的，或者说算法是好的。对于适应性的要求意味着要求设计一个能对范围广泛的问题类别提供答案的通用算法。但一般情况下，算法所处理的问题类别范围越窄，这个算法就越容易设计。要求对将来的适应性，主要是涉及算法的易修改程度，即为了让一个算法能够很容易地对付这类问题的变化。

(3) 设计的观点

标准 4　算法的设计时间最少。

从上述的各标准看，试图把算法设计的时间减少到最低限度，这会使我们所设计的算法或许不是十分有效的。但是，在某些情况下，一个问题的解答常常是在特定时间内急需的。也就是说，当制定一个高效率的算法所耗费的成本和时间可能远远超过了执行这个算法时所节省的成本和时间时，只要能设计出一个提供正确答案的算法，就可以认为是算法设计的最有效途径，亦可将由此产生的算法视为好的算法。

(4) 交流的观点

标准 5　算法最容易理解。

从交流的观点出发，使算法容易理解，将这种易理解的程序增至最大，被认为是十分重要的。算法的易理解，也为修改算法带来了方便，这也是非常重要的。

在实际的算法设计中，需要根据具体情况进行分析并加以折衷，不同的设计目标和相应的分析标准的结合才会得出有价值的结论。不同标准有时不宜在同一个算法分析中使用。在这里，我们主要以第一条标准来衡量一个算法的优劣，即