

DIANXING JISUANJI SUANFA DE
FENXI SHEJI YU SHIXIAN

典型计算机算法的 分析、设计与实现



郭红涛 著



中国水利水电出版社
www.waterpub.com.cn

TP30
471

典型计算机算法的 分析、设计与实现

郭红涛 著



中国水利水电出版社
www.waterpub.com.cn

• 北京 •

内 容 提 要

本书将典型的经典问题和算法设计技术巧妙地进行结合,系统地论述算法设计技术及其在经典问题中的应用。主要内容包括:计算机算法的基础知识、算法复杂性分析、贪心算法、分治策略、动态规划、随机算法、图的搜索算法、NP 完全问题。本书结构合理,内容丰富,深入浅出,图例丰富,理论性与实用性并重,可读性强,是一本值得学习研究的著作。

图书在版编目(CIP)数据

典型计算机算法的分析、设计与实现 / 郭红涛著
— 北京:中国水利水电出版社,2016.9
ISBN 978-7-5170-4654-7

I. ①典… II. ①郭… III. ①计算机算法 IV.
①TP301.6

中国版本图书馆CIP数据核字(2016)第203600号

责任编辑:杨庆川 陈 洁 封面设计:马静静

书 名	典型计算机算法的分析、设计与实现 DIANXING JISUANJI SUANFA DE FENXI、SHEJI YU SHIXIAN
作 者	郭红涛 著
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座100038) 网址:www.waterpub.com.cn E-mail:mchannel@263.net(万水) sales@waterpub.com.cn 电话:(010)68367658(营销中心)、82562819(万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京厚诚则铭印刷科技有限公司
印 刷	三河市佳星印装有限公司
规 格	170mm×240mm 16开本 15.5印张 201千字
版 次	2016年9月第1版 2016年9月第1次印刷
印 数	0001—1500册
定 价	46.50元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

前 言

计算机行业是个肥沃且充满勃勃生机的生态圈,不断孕育着一代又一代的新技术、新概念,毫无疑问,那些站在科技浪尖的技术概念自然成为开发者的宠儿。纵观计算机行业的发展历程,不难发现无论该行业的浪潮多么朝夕莫测,计算机和软件发展背后的根基却岿然屹立、经年不变,算法便是其根基之一,它对计算机行业的发展起着不可估量的作用。

“什么是算法?”一个常见的回答是,“完成一个任务所需的一系列步骤”。在日常生活中经常会碰到算法,刷牙的时候会执行一个算法:打开牙膏盖,拿出牙刷,持续执行挤牙膏操作直到足量的牙膏涂在你的牙刷上,盖上牙膏盖,将牙刷放到嘴的 $1/4$ 处,上下移动牙刷 N 秒,等等。如果你必须乘通勤车去工作,乘通勤车也是一个算法。诸如此类。

计算机算法与日常所运行的算法一样会影响人每天的生活。你使用过 GPS 来寻找旅行路线吗?它运行一种称为“最短路径”的算法以寻求路线。你在网上购买商品吗?那么你会使用(应该正在使用)一个运行加密算法的安全网站。当你在网上购买商品时,它们是由一个私营快递公司发货的吗?它使用算法将包裹分配给不同的卡车,然后确定每个司机发件的顺序。算法运行在各种设备上——在你的笔记本上,服务器上,智能手机上,嵌入式系统上(例如你的车中,你的微波炉中,或者气候控制系统中)——无处不在!

本书将典型的经典问题和算法设计技术巧妙地进行结合,系统地论述算法设计技术及其在经典问题中的应用。全书共 8 章。第 1 章介绍计算机算法的基础知识。第 2 章对算法的复杂性进行了分析。第 3~8 章分别介绍贪心算法、分治策略、动态

规划、随机算法、图的搜索算法、NP 完全问题。

由于时间仓促,作者水平有限,本书难免存在疏漏之处,恳请广大读者批评指正,不吝赐教。

作 者

2016 年 3 月

目 录

前言

第 1 章	计算机算法的基础知识	1
1.1	算法及其描述	1
1.2	算法与程序	6
1.3	基本的数据结构	11
第 2 章	算法复杂性分析	15
2.1	算法的时间复杂性分析	15
2.2	算法的空间复杂性分析	23
2.3	算法的渐进符号	24
2.4	算法分析实例	27
第 3 章	贪心算法	36
3.1	概述	36
3.2	背包问题	38
3.3	哈弗曼编码	44
3.4	最小生成树	53
3.5	单源最短路径	60
第 4 章	分治策略	70
4.1	概述	70
4.2	二分搜索	76
4.3	合并排序	79
4.4	快速排序	84
4.5	凸包问题	91

4.6	整数乘法	99
4.7	分析分治法在安排循环赛中的应用	101
第5章	动态规划	104
5.1	概述	104
5.2	矩阵连乘	112
5.3	多段图	113
5.4	最优路径搜索	118
5.5	最优二叉搜索树	125
5.6	最大子段和	129
5.7	图像压缩	135
5.8	电路布线	138
第6章	随机算法	141
6.1	概述	141
6.2	数值随机化算法	146
6.3	蒙特卡罗算法	150
6.4	舍伍德算法	154
6.5	拉斯维加斯算法	158
第7章	图的搜索算法	162
7.1	深度优先	162
7.2	宽度优先	169
7.3	回溯法	172
7.4	分支限界法	191
第8章	NP 完全问题	204
8.1	P 类问题	204
8.2	NP 类问题	206
8.3	NP 完全问题	207
	参考文献	240

第 1 章 计算机算法的基础知识

计算机算法是以一步接一步的方式来详细描述计算机如何将输入转化为所要求的输出的过程,或者说,算法是对计算机上执行的计算过程的具体描述。

1.1 算法及其描述

算法,简言之就是解决问题的方法。人们解决问题的过程一般由若干步骤组成,通常把解决问题的确定方法和有限步骤称为算法。如果相关问题的解决最终由计算机来实现,又由于计算机不具备思考能力以及人的“跳跃性思维”等因素,因此方法的确定和对步骤的描述尤为重要。

算法是对解题过程的描述,这种描述是建立在程序设计语言这个平台之上的。就算法的实现平台而言,可以抽象地对算法的定义如下。

算法 = 控制结构 + 原操作(对固有数据类型的操作)

无论是面向对象程序设计语言,还是面向过程的程序设计语言,都是用三种基本结构(顺序结构、选择结构和循环结构)来控制算法流程的。每个结构都应该是单入口单出口的结构体。结构化算法设计常采用自顶向下逐步求精的设计方法,因此,要描述算法首先需要有表示三个基本结构的构件,其次能方便支持自顶向下逐步求精的设计方法。

表示算法的方式主要有自然语言、流程图、盒图、PAD图、伪代码和计算机程序设计语言。

1. 自然语言

自然语言是人们日常所用的语言,如汉语、英语、德语等,使用这些语言不用专门训练,所描述的算法自然也通俗易懂。

2. 流程图

流程图是描述算法的常用工具,就简单算法的描述而言,流程图优于其他描述算法的语言。

流程图的基本组件,如图 1-1 所示。



图 1-1 流程图的基本组件

以下是流程图的 3 种基本控制结构的描述。

(1) 顺序结构

流程图的顺序结构如图 1-2 所示。

(2) 选择结构

if-then-else 型分支,如图 1-3 所示;do-case 型多分支,如图 1-4 所示。

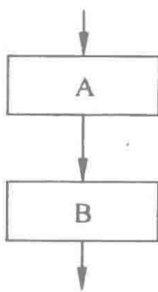


图 1-2 顺序结构

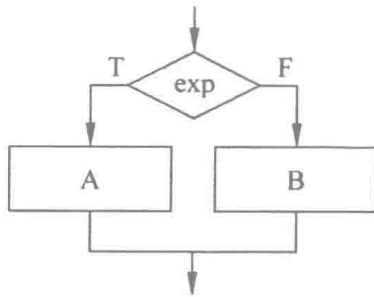


图 1-3 双分支选择结构

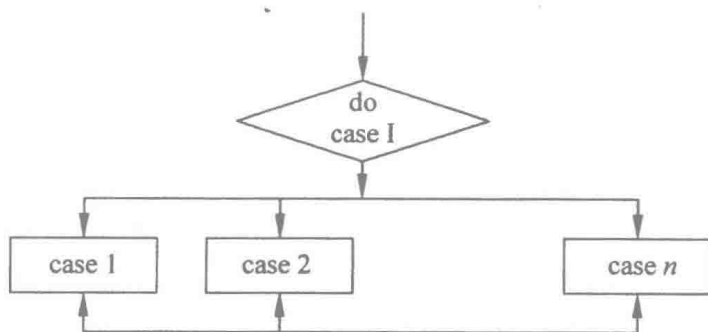


图 1-4 多分支选择结构

(3) 循环结构

do-while 型循环,如图 1-5 所示;do-until 循环结构,如图 1-6 所示。

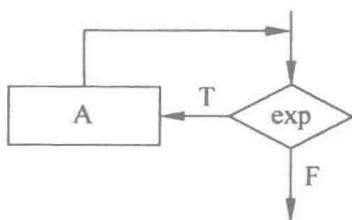


图 1-5 当型循环结构

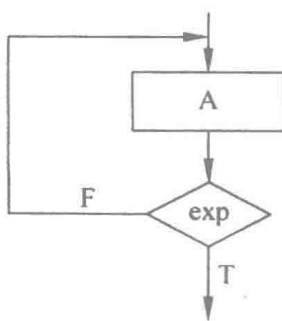


图 1-6 直到型循环结构

算法流程图虽然看起来清晰简单,但是具有一定的局限性,并没有纵观全局,所以说它并不是逐步求精的好工具。而且随意性太强,逻辑不严谨,结构化和层次感都不明显。

3. 盒图

盒图(NS 流程图)基本组件只有 3 种基本控制结构,因此能强迫算法结构化。盒图的基本控制结构可分为顺序结构、选择结构及循环结构三种。

以下是盒图的 3 种基本控制结构的描述。

(1) 顺序结构

盒图的顺序结构如图 1-7 所示。

(2) 选择结构

盒图的选择结构如图 1-8 和图 1-9 所示。

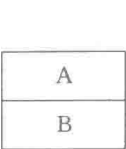


图 1-7 顺序结构

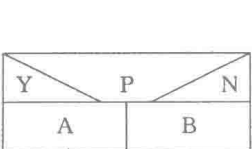


图 1-8 双分支选择结构

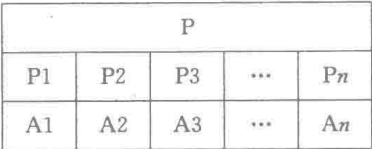


图 1-9 多分支选择结构

(3) 循环结构

盒图的循环结构如图 1-10 和图 1-11 所示。

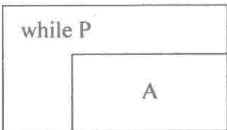


图 1-10 当型循环结构

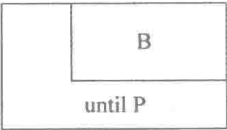


图 1-11 直到型循环结构

4. PAD 图

问题分析图(problem analysis diagram,PAD) 是一个二维树形结构图,层次感强、嵌套明确且有清晰的控制流程,综合了自然语言、流程图、盒图等算法描述方式的优点。

(1) 顺序结构

PAD 图的顺序结构如图 1-12 所示。

(2) 选择结构

PAD 图的选择结构如图 1-13 和图 1-14 所示。

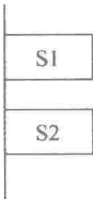


图 1-12 顺序结构

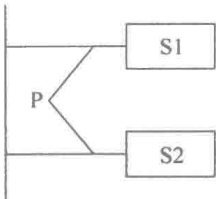


图 1-13 双分支选择结构

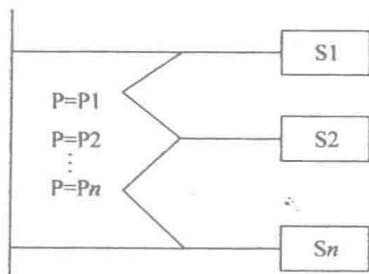


图 1-14 多分支选择结构

(3) 循环结构

PAD 图的循环结构如图 1-15 和图 1-16 所示。

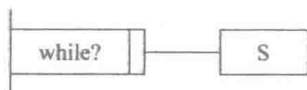


图 1-15 当型循环结构

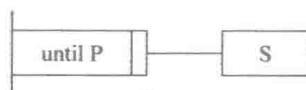


图 1-16 直到型循环结构

图 1-17 是用问题分析图描述的一个算法模块。

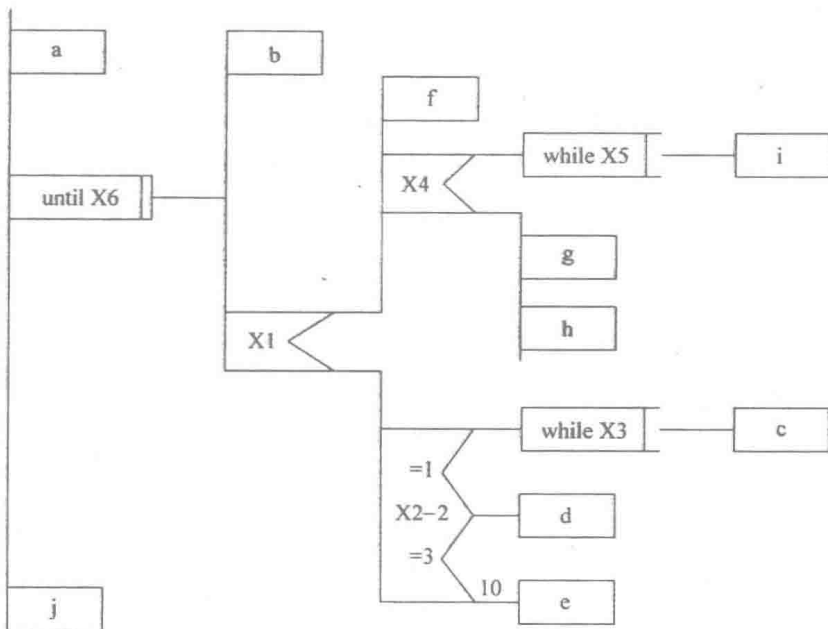


图 1-17 问题分析图实例

PAD 图的优点如下：

① 使用表示结构化控制结构的 PAD 符号设计出来的算法一定是结构化的,这一点毋庸置疑。

② 使用 PAD 图对算法进行描绘,结构清晰,一目了然。

③ 使用 PAD 图对算法进行描述,有利于用户的理解与记忆。

④ 很容易将 PAD 图转换成高级程序语言源程序,这种转换可由软件工具自动完成。

⑤ 不仅可以表示算法逻辑,还可以描绘数据结构。

⑥ PAD 图的符号支持自顶向下、逐步求精方法的使用。

PAD 图的缺点:由于 PAD 是用图形符号书写,与其他语言相比,编辑、录入操作不方便。

5. 伪代码

伪代码介于自然语言和计算机语言之间,不用图形符号,可以将整个算法运行过程的结构用接近自然语言的形式描述出来,使被描述的算法可以容易地以任何一种编程语言实现。与程序设计语言相比,使用伪代码对算法进行描述,更便于理解。

6. 程序设计语言

计算机只能识别程序设计语言,因此,使用自然语言、流程图、PAD 图、盒图、伪代码对算法进行描述最终还是要转换为计算机可以识别的程序设计语言。程序设计语言是一种被标准化的交流技巧,用来向计算机发出指令,具有其他语言无法比拟的严谨性。

1.2 算法与程序

所谓程序,就是一组计算机能识别与执行的指令。每一条指令使计算机执行特定的操作,用来完成一定的功能。

计算机的一切操作都是由程序控制的,离开了程序,计算机将一事无成。从这个意义来说,计算机的本质是程序的机器,程序是计算机的灵魂。

那么,程序与算法是什么关系呢?

算法是程序的核心。程序是某一算法用计算机程序设计语言的具体实现。事实上,当一个算法使用计算机程序设计语言描述时,就是程序。具体来说,一个算法使用C语言描述,就是C程序。

程序设计的基本目标是应用算法对问题的原始数据进行处理,从而解决问题,获得所期望的结果。在能实现问题求解的前提下,要求算法运行的时间短,占用系统空间小。

初学者往往把程序设计简单地理解为编写一个程序,这是不全面的。一个程序应包括对数据的描述与对运算操作的描述两个方面的内容。

著名计算机科学家尼克劳斯·沃思(Niklaus Wirth)就此提出一个公式:

$$\text{数据结构} + \text{算法} = \text{程序}$$

数据结构是对数据的描述,而算法是对运算操作的描述。

实际上,一个程序除了数据结构与算法这两个要素之外,还应包括程序设计方法。一个完整的C程序除了应用C语言对算法的描述之外,还包括数据结构的定义以及调用头文件的指令。

如何根据案例的具体情况确定并描述算法,如何为实现该算法设置合适的数据结构,是求解实际案例必须面对的问题。

例 1.1 构建对称方阵。

试观察图 1-18 所示的横竖折对称方阵(a)与斜折对称方阵(b)的构造特点,总结归纳其构造规律,设计并输出以上两种形式的 n (奇数) 阶对称方阵。

这是一道培养与锻炼观察能力、归纳能力与设计能力的有趣案例。

设置二维数组 $a[m][n]$ 存储 n 阶方阵的元素,数组

$a[n][n]$ 就是数据结构。本例求解算法主要是给以数组赋值与输出。一个一个元素赋值显然行不通,必须根据方阵的构造特点,归纳其构造规律,分区域给各元素赋值。

(1) 横竖折对称方阵

构造规律与赋值要点:

观察横竖折对称方阵的构造特点,方阵横向与纵向正中各有一个对称轴。两个对称轴所分4个小矩形区域表现为自对称轴向两侧递减,至4顶角元素为1。

设阶数 n (奇数) 从键盘输入,对称轴为 $m = \frac{n+1}{2}$ 。

设置二维 a 数组存储方阵行号为 i ,列号为 j , $a[i][j]$ 为第 i 行第 j 列元素。

可知主对角线(从左上至右下)有 $i = j$,次对角线(从右上至左下)有 $i + j = n + 1$ 。

按两条对角线把方阵分成上部、左部、右部与下部4个区,如图1-19所示。

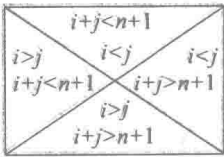
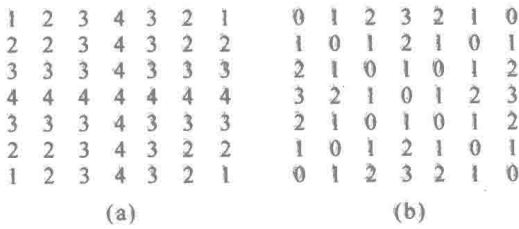


图 1-18 两个 7 阶对称方阵

图 1-19 对角线分成的 4 个区

对角线上的元素可归纳到上、下部。

上、下部按列号 j 的函数 $m - \text{abs}(m - j)$ 赋值:

if($i + j \leq n + 1 \& \& i \leq j \mid \mid i + j > n + 1 \& \& i > j$)
 $a[i][j] = m - \text{abs}(m - j);$

左、右部按行号 i 的函数 $m - \text{abs}(m - i)$ 赋值:

if($i + j < n + 1 \& \& i > j \mid \mid i + j > n + 1 \& \& i < j$)
 $a[i][j] = m - \text{abs}(m - i);$

程序设计:

```

//横竖折对称方阵, c141
#include<stdio.h> //调用两个头文件
#include<math.h>
void main()
{ int i,j,m,n,a[30][30]; //定义数据结构
  printf("请确定方阵阶数(奇数)n:");scanf("%d",&n);
  if(n%2==0)
  {
    printf("请输入奇数!");
    return;
  }
  m=(n+1)/2;
  for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
    {
      if(i+j<=n+1&&i<=j||i+j>=n+1&&i>=j)
        a[i][j]=m-abs(m-j); //方阵上、下部元素赋值
      if(i+j<=n+1&&i>=j||i+j>=n+1&&i<=j)
        a[i][j]=m-abs(m-i); //方阵左、右部元素赋值
    }
  printf("%d阶对称方阵为:\n",n);
  for(i=1; i<=n; i++)
  { for(j=1;j<=n;j++) //输出对称方阵
    printf("%3d",a[i][j]);
    printf("\n");
  }
}

```

(2) 斜折对称方阵构造规律与赋值要点

斜折对称方阵的两个对角线上均为 0, 依两对角线把方阵分为 4 个区域, 每一区域表现为同数字依附两对角线折叠对称, 至上下左右正中元素为 $\frac{n}{2}$ 。

同样设置二维 $a[n][n]$ 数组存储方阵中的元素, 行号为 i , 列号为 j , $a[i][j]$ 为第 i 行第 j 列元素。

令 $m = \frac{n+1}{2}$, 按 m 把方阵分成的 4 个小矩形区, 如图 1-20 所示。

注意到方阵的主对角线(从左上至右下)上的元素为 $i = j$, 则左上区与右下区依主对角线赋值:

$$a[i][j] = \text{abs}(i - j);$$

注意到方阵的次对角线(从右上至左下)上的元素为 $i + j = n + 1$, 则右上区与左下区依次对角线赋值:

$$a[i][j] = \text{abs}(i + j - n - 1);$$

程序设计:

```
//斜折对称方阵, c142
#include<math.h>
#include<stdio.h>
void main()
{
    int i,j,m,n,a[30][30];
    printf("请确定方阵阶数(奇数)n:");scanf("%d",&n),
    if(n%2==0)
    {
        printf("请输入奇数!");
        return;
    }
    m=(n+1)/2;
    for(i=1;i<=n;i++)
        for(j=1;j<=n;j++)
        {
            if(i<=m&&j<=m||i>m&&j>m)
                a[i][j]=abs(i-j);           //方阵左上部与右下部元素赋值
            if(i<=m&&j>m||i>m&&j<=m)
                a[i][j]=abs(i+j-n-1);       //方阵右上部与左下部元素赋值
        }
    printf("%d阶对称方阵为:\n",n);
    for(i=1;i<=n;i++)
    {
        for(j=1;j<=n;j++)
            printf("%3d",a[i][j]);         //输出对称方阵
        printf("\n");
    }
}
```

以上两个完整的 C 程序包含了算法描述(整数的输入、数组元素的赋值与输出)、数据结构(a 数组与变量 i, j, m, n) 的定义以及两个 C 头文件的调用。

运行以上两个程序,可以在欣赏各个具体的对称方阵中感受从特别到一般的神奇。

$i \leq m$ $j \leq m$	$i \leq m$ $j > m$
$i > m$ $j \leq m$	$i > m$ $j > m$

图 1-20 按 m 分成的 4 个小矩形区