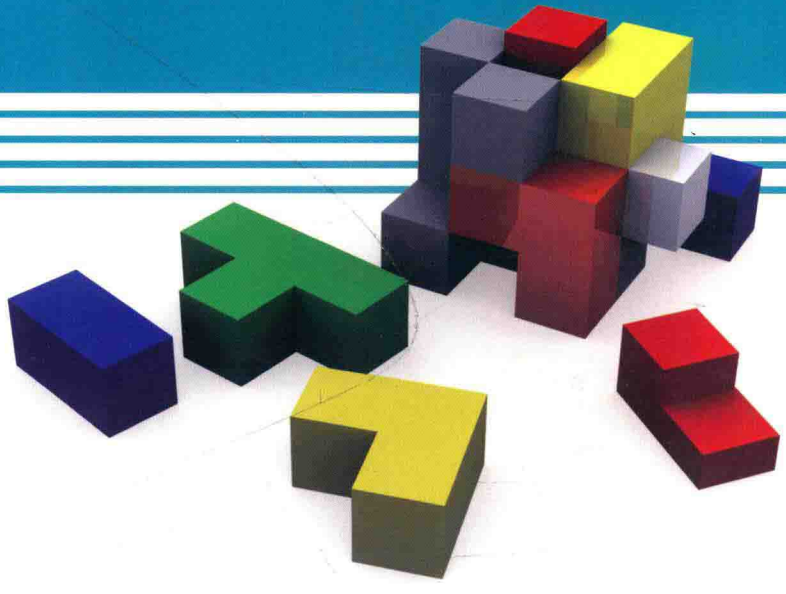
高等学校数据结构课程系列教材

算法设计与分析（第2版）

◎ 李春葆　主编



微课版
20小时
教学视频

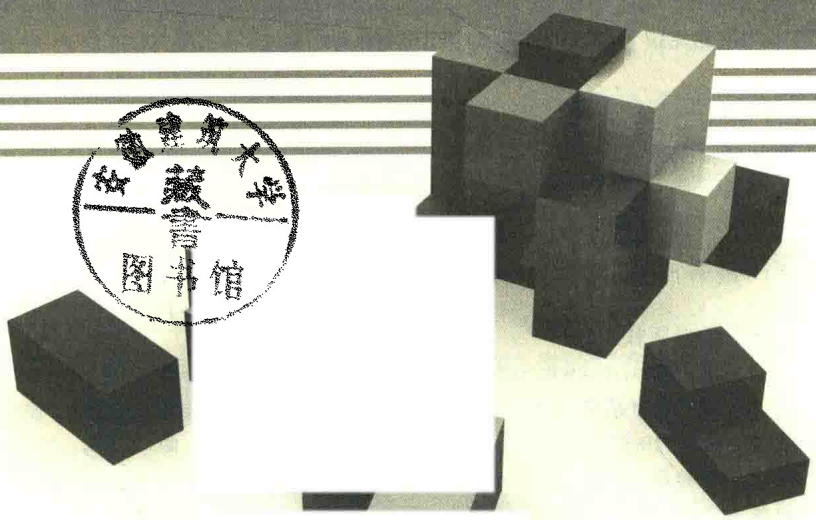
- 循序渐进，强调算法设计和动手能力培养
- 注重实验，提供大量的上机实验题和在线编程题
- 案例来源于著名IT企业面试笔试题和ACM竞赛题
- 教学资源丰富，每个知识点都配套了视频讲解

清华大学出版社

居结构课程系列教材

算法设计与分析
（第2版）

◎ 李春葆 主编
李筱驰 蒋林 陈良臣 喻丹丹 编著



清华大学出版社
北京

内 容 简 介

本书系统地介绍了各种常用的算法设计策略，包括递归、分治法、蛮力法、回溯法、分枝限界法、贪心法、动态规划、概率算法和近似算法等，并详细讨论了各种图算法和计算几何设计算法。

全书既注重原理又注重实践，配有大量图表、练习题、上机实验题和在线编程题，内容丰富，概念讲解清楚，表达严谨，逻辑性强，语言精练，可读性好。

本书既便于教师课堂讲授，又便于自学者阅读，适合作为高等院校“算法设计与分析”课程的教材，也可供 ACM 和各类程序设计竞赛者参考。

图书在版编目(CIP)数据

算法设计与分析/李春葆主编. —2 版. —北京：清华大学出版社，2018
(高等学校数据结构课程系列教材)
ISBN 978-7-302-50098-8

Ⅰ. ①算… Ⅱ. ①李… Ⅲ. ①电子计算机－算法设计－高等学校－教材 ②电子计算机－算法分析－高等学校－教材 Ⅳ. ①TP301.6

中国版本图书馆 CIP 数据核字(2018)第 102226 号

责任编辑：魏江江　王冰飞
封面设计：刘　键
责任校对：梁　毅
责任印制：李红英

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn,010-62795954
印 装 者：三河市国英印务有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：28.75　　**字　　数**：699 千字
版　　次：2015 年 5 月第 1 版　2018 年 8 月第 2 版　　**印　　次**：2018 年 8 月第1次印刷
印　　数：8001～10000
定　　价：59.50 元

产品编号：079388-01

前　言

算法在计算科学中扮演着重要角色。算法设计是计算机科学与技术专业的必修课，其目标是培养学生分析问题和解决问题的能力，使学生掌握算法设计的基本技巧和方法，熟悉算法分析的基本技术，并能熟练运用一些常用算法策略解决一些较综合的问题。

在学习本书之前，学生已经学习了基本的数据结构知识，能熟练运用一门或多门编程语言，并具备了一定的编程经验。如何利用已学过的知识针对不同的实际问题设计出有效的算法，是本书所要达到的目的。

本书的特点是“问题模型化，求解算法化，设计最优化”，在掌握必要的算法设计技术和编程技巧的基础上，能够在实际工作中根据具体问题设计和优化算法。本书是针对这一特点并结合课程组全体教师多年的教学经验编写的。

1. 本书内容

全书由 12 章构成，各章的内容如下。

第 1 章　概论：介绍算法的概念、算法分析方法和 STL 在算法设计中的应用。

第 2 章　递归算法设计技术：介绍递归的概念、递归算法设计方法和相关示例、递归算法到非递归算法的转化以及递推式的计算。

第 3 章　分治法：介绍分治法的策略和求解过程，讨论采用分治法求解排序问题、查找问题、最大连续子序列和问题、大整数乘法问题及矩阵乘法问题的典型算法，并简要介绍了并行计算的概念。

第 4 章　蛮力法：介绍蛮力法的特点、蛮力法的基本应用示例、递归在蛮力法中的应用以及图的深度优先和广度优先遍历算法。

第 5 章　回溯法：介绍解空间概念和回溯法算法框架，讨论采用回溯法求解 0/1 背包问题、装载问题、子集和问题、n 皇后问题、图的 m 着色问题、任务分配问题、活动安排问题和流水作业调度问题的典型算法。

第 6 章　分枝限界法：介绍分枝限界法的特点和算法框架、队列式分枝限界法和优先队列式分枝限界法，讨论采用分枝限界法求解 0/1 背包问题、图的单源最短路径、任务分配问题和流水作业调度问题的典型算法。

第 7 章　贪心法：介绍贪心法的策略、求解过程和贪心法求解问题应具有的性质，讨论采用贪心法求解活动安排问题、背包问题、最优装载问题、田忌赛马问题、多机调度问题、哈夫曼编码和流水作业调度问题的典型算法。

第 8 章　动态规划：介绍动态规划的原理和求解步骤，讨论采用动态规划法求解整数拆分问题、最大连续子序列和问题、三角形最小路径问题、最长公共子序列问题、最长递增子序列问题、编辑距离问题、0/1 背包问题、完全背包问题、资源分配问题、会议安排问题和滚

动数组的典型算法。

第9章 图算法设计：讨论构造图最小生成树的两种算法(Prim和Kruskal算法，并查集的应用)、求图的最短路径的4种算法(Dijkstra、Bellman-Ford、SPFA和Floyd)，并采用5种算法策略求解旅行商问题(TSP问题)，最后介绍网络流的相关概念以及求最大流和最小费用最大流的算法。

第10章 计算几何：介绍计算几何中常用的矢量运算以及求解凸包问题、最近点对问题和最远点对问题的典型算法。

第11章 计算复杂性理论简介：介绍图灵机计算模型、P类和NP类问题以及NPC问题。

第12章 概率算法和近似算法：介绍这两类算法的特点和基本的算法设计方法。

书中带“*”符号的章节作为选学内容。

2. 本书特色

本书具有如下鲜明特色。

(1) 由浅入深，循序渐进：每种算法策略从设计思想、算法框架入手，由易到难地讲解经典问题的求解过程，使读者既能学到求解问题的方法，又能通过对算法策略的反复应用掌握其核心原理，以收到融会贯通之效。

(2) 示例丰富，重视启发：书中列举大量的具有典型性的求解问题，深入剖析采用相关算法策略求解的思路，展示算法设计的清晰过程，并举一反三，激发学生学习算法设计的兴趣。

(3) 注重求解问题的多维性：同一个问题采用多种算法策略实现，如0/1背包问题采用回溯法、分枝限界法和动态规划求解，旅行商问题采用5种算法策略求解。通过不同算法策略的比较，使学生更容易体会到每一种算法策略的设计特点和各自的优/缺点，以提高算法设计的效率。

(4) 强调实验和动手能力的培养：算法讲解不仅包含思路描述，而且以C/C++完整程序的形式呈现，同时给出了大量的上机实验题和在线编程题，大部分是近几年国内外的著名IT企业面试笔试题(谷歌、微软、阿里巴巴、腾讯、网易等)和ACM竞赛题。通过这些题目的训练，不仅可以提高学生的编程能力，而且可以帮助其直面求职市场。

(5) 本书配套有《算法设计与分析(第2版)学习与实验指导》(李春葆，清华大学出版社，2018)，涵盖所有练习题、上机实验题和在线编程题的参考答案。

(6) 本书配套有绝大部分知识点的教学视频，视频采用微课碎片化形式组织(含100多个小视频，累计超过20小时)，读者通过扫描二维码即可观看相关视频讲解。

3. 教学资源

本书提供的教学资源包括完整的教学PPT和书中全部源程序代码(在VC++6.0中调试通过)，用户可以扫描封底课件二维码免费下载。

4. 感谢

本书的编写得到湖北省教育厅和武汉大学教学研究项目《计算机科学与技术专业课程体系改革》的大力帮助，清华大学出版社的魏江江主任全力支持本书的编写工作，王冰飞老

师给予精心的编辑工作。

本书在编写过程中参考了很多同行的教材和网络博客，特别是“牛客网”中众多的企业面试、笔试题和丰富资源给予编者良好的启发，河南工程学院张天伍老师和使用本教材第1版的多位老师指正多处问题和错误，在此一并表示衷心感谢。

本书是课程组全体教师多年教学经验的总结和体现，尽管编者不遗余力，但由于水平所限，难免存在不足之处，敬请教师和同学们批评指正，在此表示衷心感谢。

编　者

2018年5月

目　录

第 4 章 蛮力法 /120

第 5 章 回溯法 /163

第1章 概论

算法是程序的灵魂，一个程序应包括对数据的表示(数据结构)和对操作的描述(算法)两个方面的内容，所以著名计算机科学家沃思提出了“数据结构＋算法＝程序”的概念。同一问题可能有多种求解算法，通过算法时间复杂度和空间复杂度分析判定算法的好坏。本章讨论算法设计和分析的相关概念，以及采用C++标准模板库(STL)设计算法的方法。

1.1 算法的概念

1.1.1 什么是算法

算法(algorithm)是求解问题的一系列计算步骤，用来将输入数据转换成输出结果，如图1.1所示。如果一个算法对其每一个输入实例都能输出正确的结果并停止，则称它是正确的。一个正确的算法解决了给定的求解问题，不正确的算法对于某些输入来说可能根本不会停止，或者停止时给出的不是预期的结果。

图1.1 算法的概念

算法设计应满足以下几个目标。

扫一扫

视频讲解

(1) 正确性：要求算法能够正确地执行预先规定的功能和性能要求。这是最重要也是最基本的标准。

(2) 可使用性：要求算法能够很方便地使用。这个特性也叫作用户友好性。

(3) 可读性：算法应该易于人的理解，也就是可读性好。为了达到这个要求，算法的逻辑必须是清晰的、简单的和结构化的。

(4) 健壮性：要求算法具有很好的容错性，即提供异常处理，能够对不合理的数据进行检查，不经常出现异常中断或死机现象。

(5) 高效率与低存储量需求：通常，算法的效率主要指算法的执行时间。对于同一个问题如果有多种算法可以求解，执行时间短的算法效率高。算法存储量指的是算法执行过程中所需的最大存储空间。效率和低存储量都与问题的规模有关。

【例1.1】 以下算法用于在带头结点的单链表 h 中查找第一个值为 x 的结点，找到后返回其逻辑序号(从1计起)，否则返回0。分析该算法存在的问题。

```
#include <stdio.h>
typedef struct node
{   int data;
    struct node *next;
} LNode;                                    //单链表结点类型定义
int findx(LNode *h,int x)
{   LNode *p=h->next;
    int i=0;
    while (p->data!=x)
    {   i++;
        p=p->next;
    }
```

```
    return i;
}
```

解 当单链表中的首结点值为 x 时该算法返回 0，此时应该返回逻辑序号 1。另外，当单链表中不存在值为 x 的结点时该算法执行出错，因为 p 为 NULL 时仍执行 p=p->next。所以该算法不满足正确性和健壮性，应改为如下算法。

```
int findx(LNode *h,int x)
{   LNode *p=h->next;                    //p 初始时指向首结点
    int i=1;
    while (p!=NULL && p->data!=x)
    {   i++;
        p=p->next;
    }
    if (p==NULL)                         //没找到值为 x 的结点返回 0
        return 0;
    else                                 //找到值为 x 的结点返回其逻辑序号 i
        return i;
}
```

算法具有以下 5 个重要特性。

(1) 有限性：一个算法必须总是(对任何合法的输入值)在执行有限步之后结束，且每一步都可在有限时间内完成。

(2) 确定性：算法中的每一条指令必须有确切的含义，不会产生二义性。

(3) 可行性：算法中的每一条运算都必须是足够基本的，也就是说它们在原则上都能精确地执行，甚至人们仅用笔和纸做有限次运算就能完成。

(4) 输入性：一个算法有零个或多个输入。大多数算法的输入参数是必要的，但对于较简单的算法，如计算 1+2 的值，不需要任何输入参数，因此算法的输入可以是零个。

(5) 输出性：一个算法有一个或多个输出。算法用于某种数据处理，如果没有输出，这样的算法是没有意义的，这些输出是和输入有着某些特定关系的量。

说明：算法和程序是有区别的，程序是指使用某种计算机语言对一个算法的具体实现，即具体要怎么做，而算法侧重于对解决问题的方法描述，即要做什么。算法必须满足有限性，而程序不一定满足有限性，例如 Windows 操作系统在用户没有退出、硬件不出现故障以及不断电的条件下理论上可以无限时运行，所以严格上讲算法和程序是两个不同的概念。当然，算法也可以直接用计算机程序来描述，本书就是采用这种方式。

【例 1.2】 有下列两段描述。

描述 1：

```
void exam1()
{   int n;
    n=2;
    while (n%2==0)
        n=n+2;
    printf("%d\n",n);
}
```

描述 2：

```
void exam2()
{   int x,y;
    y=0;
    x=5/y;
    printf("%d,%d\n",x,y);
}
```

这两段描述均不能满足算法的特性，它们违反了算法的哪些特性？

解 第一段是一个死循环，违反了算法的有限性特性；第二段出现除零错误，违反了算法的可行性特性。

1.1.2 算法描述

描述算法的方式很多，有的采用类 Pascal 语言，有的采用自然语言伪码。本书采用 C/C++语言描述算法的实现过程，通常用 C/C++函数来描述算法。

以设计求 $1+2+\cdots+n$ 的值的算法为例说明 C/C++语言描述算法的一般形式，该算法如图 1.2 所示。

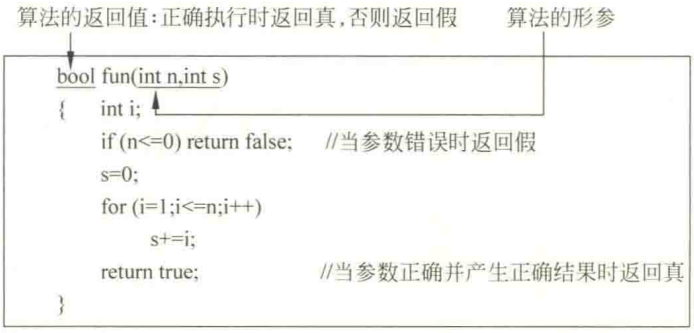


图 1.2 算法描述的一般形式

通常用函数的返回值表示算法能否正确执行，当算法只有一个返回值或者返回值可以区分算法是否正确执行时用函数返回值来表示算法的执行结果，另外还可以带有形参表示算法的输入/输出。任何算法（用函数描述）都是被调用的（在 C/C++语言中除 main 函数外任何一个函数都会被其他函数调用，如果一个函数不被调用，这样的函数是没有意义的）。在 C 语言中调用函数时只有从实参到形参的单向值传递，在执行函数时若改变了形参，对应的实参不会同步改变。例如设计以下主函数调用上面的 fun 函数。

```
void main()
{   int a=10,b=0;
    if (fun(a,b)) printf("%d\n",b);
    else printf("参数错误\n");
}
```

在执行时发现输出结果为 0，因为 b 对应的形参为 s，fun 执行后 $s=55$，但 s 并没有回传给实参 b。在 C 语言中可以用传指针方式来实现形参的回传，但增加了函数的复杂性。为此在 C++语言中增加了引用型参数的概念，引用型参数名前需加上 &，表示这样的形参在执行后会将结果回传给对应的实参。上例采用 C++语言描述算法如图 1.3 所示。

当将形参 s 改为引用类型的参数后，在执行时 main 函数的输出结果就正确了，即输出 55。由于 C 语言不支持引用类型，C++语言支持引用类型，所以本书的算法描述语言为 C/C++语言。需要注意的是，在 C/C++语言中数组本身就是一种引用类型，所以当数组作为形参需要回传数据时其数组名之前不需要加 &，它自动将形参数组的值回传给实参数组。