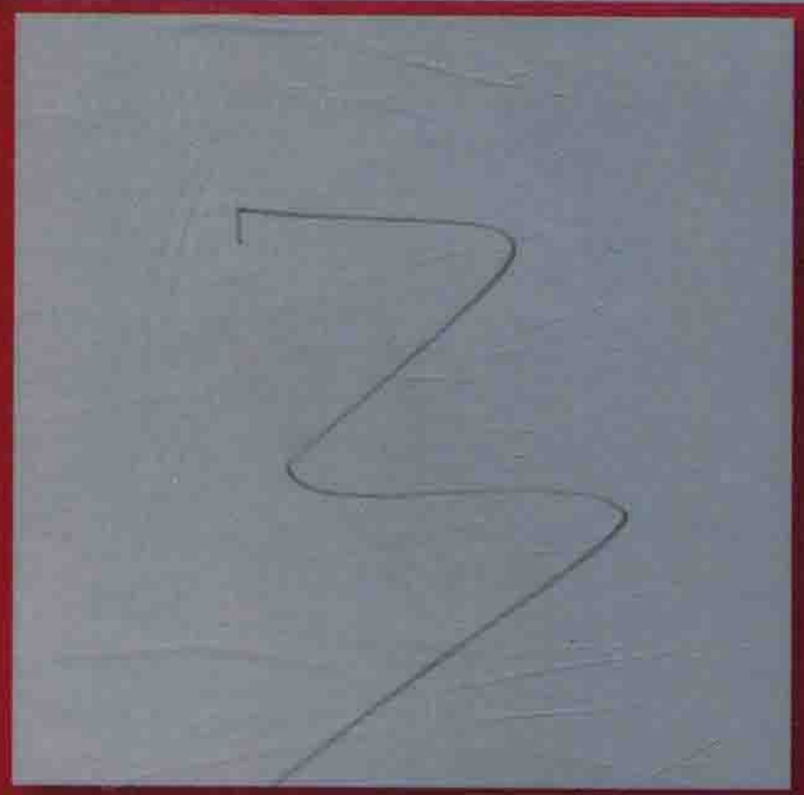
21世纪高等学校计算机专业实用规划教材

Design and Analysis of Algorithm Using Python

算法设计与分析（Python）

程振波 李曲 王春平 编著



清华大学出版社

21世纪高等学校计算机专业实用规划教材

Design and Analysis of Algorithm Using Python

算法设计与分析
（Python）

程振波 李曲 王春平 编著

清华大学出版社
北京

内 容 简 介

本书介绍了算法设计与分析的基本技巧，主要包括递归、分治、动态规划、贪心和随机等算法，以及利用这些算法求解计算问题的时间复杂度分析等内容。通过诸多有趣的实例，向读者介绍了算法设计的思想，以便读者能形成算法思维的固定模式去解决问题。在介绍每一类算法范式以及分析算法复杂度时，都力求建立直观的思维过程，而摒弃过深的数学证明。书中所有算法均采用 Python 语言描述，读者能从中学习到许多算法实现的技巧，从而提高编写程序的能力。

本书可作为高等学校计算机专业大一、大二或者学习过程序设计的非计算机专业学生的算法设计与分析教材。

图书在版编目(CIP)数据

算法设计与分析：Python/程振波，李曲，王春平编著.—北京：清华大学出版社，2018
(21 世纪高等学校计算机专业实用规划教材)
ISBN 978-7-302-47748-8

Ⅰ.①算… Ⅱ.①程… ②李… ③王… 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(2017)第 166911 号

责任编辑：梁　颖　柴文强
封面设计：何凤霞
责任校对：焦丽丽
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 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　印　张：15　　字　数：368 千字
版　　次：2018 年 1 月第 1 版　　印　次：2018 年 1 月第 1 次印刷
印　　数：1～2000
定　　价：39.00 元

产品编号：069618-01

前　言

“算法设计与分析”是计算机专业非常重要的一门基础课程，它不仅是诸多计算机专业课程的基础，也是许多信息科技类公司招聘程序员时，笔试与面试重点考核的内容。算法设计与分析已经有了诸多经典的著作，比如美国麻省理工学院（MIT）几位教授合著的《算法导论》[①] 等。然而，这些经典著作当作教材使用时，都会存在对内容进行适当裁剪，以便更适合 48 或者 32 个学时教学的问题。我们写本书的目的就是对初等算法内容进行合理的编排，让初学者能很快地掌握解决计算问题的常用算法，以及分析算法效率的方法。

本书算法均采用 Python 语言进行描述，Python 是一类解释性语言，其语法简单直观，有一定程序设计基础的学生可以很快入门。Python 语法简单并不意味着功能弱，它在科学计算、Web 应用等诸多领域都有着广泛的应用。国外知名的高校，如麻省理工学院，也在算法设计课中采用 Python 语言描述。与采用伪代码描述算法的书比较而言，采用 Python 描述算法能给读者直接的运算结果，从而可以使读者更易于揣摩算法实现的技巧。

计算机算法不仅涉及诸多理论，还有各种技术细节。比如介绍随机算法时，有些执行时间的分析就需要较多的概率论知识；而算法实现技术细节则不仅关注如何存储数据，甚至对执行算法的硬件环境也会考虑在内。本书的内容安排则介于两者之间，在数学分析与实现之间期望取得合理的平衡。首先，在分析算法效率时尽量避免过深的数学证明，但关键步骤依然会给出直观的解释。其次，在实现算法时本书尽量利用 Python 已有的数据结构和库函数，从而简化算法实现的技术难度。

如果将要处理的数据、问题看作是食材，那么算法就是将食材“转化”成各种令人垂涎的美食的过程。中国菜肴到处都是充满想象力的转化，将原本普通的食材（如大豆和糯米等）转化为营养和美味的食物（豆腐、酒酿和酱料等）。本书的主线就是转化，它不仅有问题的转化，也有方法的转化（如图 1 所示）。通过问题的转化将问题“化繁为简”，通过方法的转化以便融会贯通各种算法设计的技巧。

本书主要内容

由于计算机已经成为现代科技、生活不可缺少的工具。因此，解决计算问题的算法涉及的内容可以说包罗万象，从简单的排序和查找到复杂的语音识别、文字翻译，甚至

① 见参考文献 [1]。

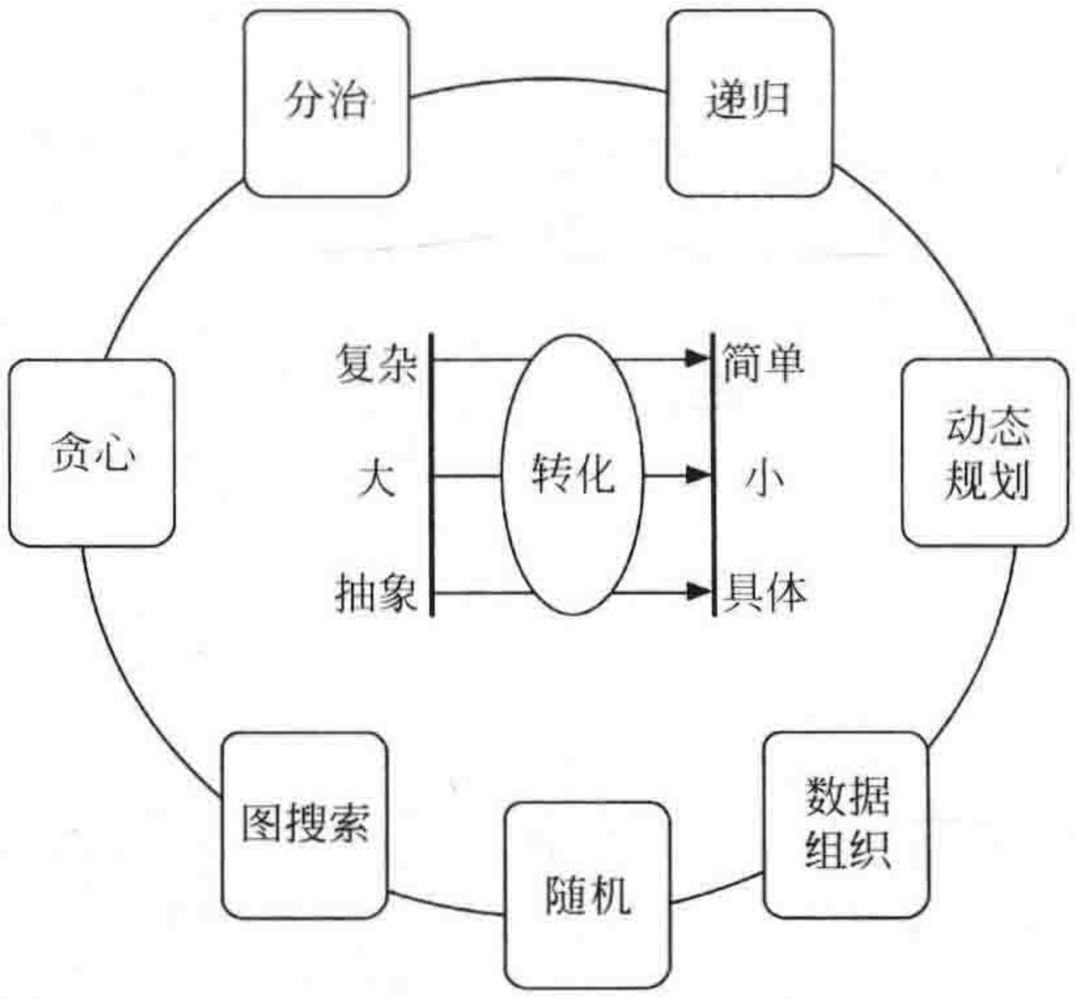


图 1 本书的主线 —— 转化

游戏等都离不开算法。本书内容涵盖了大部分的经典算法，主要内容包括递归算法、分治算法、排序算法、动态规划算法、图搜索算法、最大流算法、随机算法和算法复杂度。

第 1 章主要介绍算法的基本概念，通过实例向读者展示解决同一问题的不同算法的确存在效率上显著的差异。第 2 章介绍度量算法效率的记号，以及分析简单函数执行时间的常用技巧。第 3 章通过解决文档比较、单词拼写纠正和稳定匹配这三个有趣的问题，帮助读者熟悉 Python 语言。第 4 章介绍了递归算法以及递归函数求解，从而为后续章节复杂的算法设计与分析打下一定的基础。第 5 章介绍了组织数据的两个常用方法：排序和数据结构，主要强调递归在组织数据中的应用，帮助读者进一步熟悉采用递归求解问题的过程。

第 6 章到第 11 章则分别介绍了分治算法、图搜索、贪心、动态规划、最大流和随机算法。通过各种有趣的问题，向读者展示转化的基本技巧，以便更好地帮助读者建立采用算法思维去解决问题的习惯。第 12 章介绍了算法复杂度，帮助读者明确哪类问题“可解”，而哪类问题目前“不可解”。

本书由程振波总体设计和规划。第 2 到第 12 章由程振波编写，第 1 章由程振波、李曲和王春平编写。全书由程振波统稿。

如何使用本书

本书的内容框架是笔者在浙江工业大学“算法设计与分析”课程的讲义，内容的编排则参考了 MIT 的算法课程 6.006。[①] 因此，本书从内容安排来说非常适合学时为 48 或者 32 学时的算法课程。对于教师而言，可以直接按照本书的章节安排教学计划。为了便于教师安排教学，具体的教学建议如下：

① MIT 将“算法设计与分析”课程分解成了两门课。一门是 6.006，该课程主要是算法的入门课程，可以面向各个专业开设。另一门则是 6.046，这是一门面向有一定算法基础的学生开设的算法课程。

教学内容	学习要点及教学内容	课时安排
第 1 章　引言	• 掌握算法的定义。了解算法的来源，理解现实生活中解决问题的办法与算法之间的关系；掌握衡量算法的属性，尤其是正确性和时间效率对算法的意义。 • 掌握算法效率的基本概念。理解直接计算某个输入规模的时间来衡量算法效率的不足；了解渐进分析法以及多项式时间复杂度与指数时间复杂度的区别。 • 了解求解问题可能存在效率不同的算法。掌握求解一维高点问题的简单算法及改进算法。 • 掌握哈希表的基本概念。	2
第 2 章　渐进分析与 Python 计算模型	• 掌握运行算法的简化模型。了解单处理器随机访问机器模型的结构，以及运行在该机器模型上常见指令的执行时间。 • 掌握算法渐进分析的概念。熟悉三种渐进函数的定义，以及常见函数的渐进表示。 • 熟练掌握基本函数的渐进分析。熟悉 Python 的判断、循环语句写法，熟练掌握 Python 的基本数据结构的使用。掌握较为复杂的函数的时间复杂度分析，如求最大值、二分搜索等。	2
第 3 章　问题求解与代码优化	• 基本掌握使用 Python 求解较为复杂的问题。 • 了解文档比较问题及其算法。 • 了解单词拼写问题及其实现算法。 • 了解稳定匹配问题及其实现算法。	2
第 4 章　递归算法与递归函数	• 熟悉递归的组成结构。熟练掌握递归算法的两个基本组成，以及它们各自的作用。 • 掌握递归算法执行的过程。了解递归算法在机器模型中的运行过程。 • 熟练掌握常见问题的递归求解方法。熟悉回文、全排列和汉诺塔问题的递归算法。 • 熟练掌握求解标准递归函数 $T(n) = aT(n/b) + f(n)$ 的方法。掌握替换法和主分析法求解递归函数的过程，理解主分析法的三类条件及其对应的解。	4 或 6

续表

教学内容	学习要点及教学内容	课时安排
第 5 章　排序与树结构	• 熟悉插入排序、选择排序和合并排序算法。能熟练写出这三个排序算法的实现代码以及它们各自的时间复杂度。 • 掌握二叉搜索树的基本数据操作。能从使用场景的角度理解二叉树与数组、链表等数据结构的不同。掌握基于二叉搜索树常见的数据操作，比如插入、删除和查找等。 • 熟练掌握堆结构的应用场景和数据操作。熟悉建堆算法及其时间复杂度分析，了解基于堆的排序和合并 k 个有序序列等应用。	4
第 6 章　分治算法	• 掌握分治算法求解问题的三个基本步骤。 • 掌握利用分治法求解一些典型问题，如序列最大差值区间、统计逆序数、空间点最小距离和序列中第 k 小的数等问题。 • 熟悉如何将问题进行分解，以及合并子问题解的常用技巧。掌握分治算法的时间复杂度分析。	6
第 7 章　图搜索算法	• 熟悉图的两种常见表示方法，熟练掌握如何在计算机中存储图。了解图在计算机应用领域常见的应用场景。 • 熟练掌握图上宽度优先搜索算法及其算法复杂度分析，了解利用宽度优先搜索解决计算问题的建模过程。 • 熟练掌握图上深度优先搜索算法及其算法复杂度分析，了解利用深度优先搜索解决计算问题的建模过程。	4
第 8 章　贪心算法	• 了解贪心算法求解优化问题的过程。 • 熟练掌握利用贪心算法求解典型的计算问题，如硬币找零、间隔任务规划等问题。了解利用替换法证明贪心策略是否能获得全局最优解的过程。 • 熟练掌握贪心算法在两个典型图搜索中的应用，即单源最短路径和最小生成树。理解单源最短路径和最小生成树算法中，利用合理的数据结构优化算法复杂度的技巧。	4

续表

教学内容	学习要点及教学内容	课时安排
第 9 章　动态规划算法	• 理解动态规划求解优化问题的典型步骤，以及动态规划算法求解计算问题的时间复杂度分析。 • 熟练掌握利用动态规划算法求解一维、二维等典型优化问题，如斐波那契数、拾捡硬币、连续子序列的最大值、矩阵的括号、0-1 背包问题等。 • 对于简单问题能画出其动态规划表，并能从中得到问题的解。	6
第 10 章　最大流算法	• 掌握最大流问题的定义，了解流量、容量以及它们之间的关系。 • 掌握通过增广路径求最大流问题的 Ford-Fulkerson 和 Edmond-Karp 算法，理解这两个算法之间的异同。 • 了解将计算问题转化为最大流问题的基本过程。掌握通过最大流算法求解二向图最大匹配和文件传输中的不重合边等问题的方法。	2
第 11 章　随机算法	• 了解两种典型的随机算法：蒙特卡洛和拉斯维加斯算法，以及它们之间的异同。 • 熟练掌握利用随机算法求解典型的计算问题，如矩阵乘积结果验证、快速排序、选择第 k 小的数和最小割验证等。 • 了解随机快速排序算法复杂度分析过程。	2
第 12 章　算法复杂度	• 了解如何根据问题求解的难易程度对计算问题进行基本分类。 • 理解 P 问题、NP 问题和 NPC 问题的定义。 • 了解几个典型的 NPC 问题，理解为什么证明 P 是否等于 NP 是计算机领域最为重要的问题之一。	2

对学生而言，先阅读书中各章节内容，然后运行书中代码以便检验对算法的理解程度。特别是，学生还应该独立重复出书中各个问题的算法，这个过程就好比学习围棋的选手进行复盘一样。如果仅仅是了解算法原理，而没有通过写代码来实现算法，将不利于读者培养独立解决问题的能力。

此外，除了课后习题外，我们还建议学生在 leetcode[①] 上刷题。leetcode 上的题目

① https://leetcode.com/

不是国际计算机学会（ACM）的竞赛题目，而是各大 IT 企业的面试题目。通过解题不仅能提高学生算法设计的能力，也对编程能力有极大提高。

阅读本书需要学生能按照教程（http://www.python.org/）配置 Python 环境，知道如何写一个简单的包括循环的函数。因此，该课程安排在学生上过一门程序语言课程之后较为合适。

致谢

在本书编写过程中，浙江工业大学的许多同学阅读了初稿，尤其感谢李轶、陈明明、严凡等同学给出的许多建议。我们的许多同事也对本书提出了诸多宝贵建议，他们是吕慧强、钱能和黄德才老师。本书还受到浙江工业大学校级重点教材资助，特此感谢。对在本书出版过程中付出辛勤劳动的清华大学出版社的编辑致以特别的谢意。最后，作者程振波要对他的妻子王玉秀、女儿程静萱致以特别的谢意，感谢她们给予的爱和支持，让他能心无旁骛地完成书稿。

程振波　李　曲　王春平

czb/liqu/wangcp@zjut.edu.cn

2017 年 6 月

目　录

第1章　引　　言

本章学习目标

- 掌握算法的定义
- 掌握算法效率的基本概念
- 了解求解问题可能存在效率不同的算法

1.1　算法的定义

算法的英文 Algorithm来自于一位叫 al-Khwārizmī的波斯数学家。他在大约公元前825 年，写了一本叫*On the Calculation with Hindu Numerals*的书，该书主要列举了加、减、乘、除和计算圆周率数值的计算方法。该书后被翻译成拉丁文*Algoritmi de numero Indorum*，英文的 Algorithm 就来自拉丁文 Algoritmi。

什么是算法？简单地说，算法就是按照一定步骤解决问题的办法。这个定义里面蕴含了算法的两个重要属性：一个属性是，算法一般包括一系列有限的步骤，这些步骤能**快速**完成；另一个属性是，算法要能**正确**地给出具体问题的解。

“民以食为天”，下面通过一个与我们日常生活息息相关的例子来说明算法的这两个属性。红烧肉是中国一道非常经典的家常菜肴，烹制红烧肉的过程就可以总结为一个算法。这个算法的输入是五花肉和各种调料，如葱、姜、蒜、酱油和糖等，输出当然就是可口诱人的红烧肉（如图 1.1）。根据输入，烹制这道菜肴的步骤包括：

- 五花肉洗净，切 4cm 见方的块备用
- 葱切大段、姜切片、蒜剥好备用
- 用纱布将大料、花椒、桂皮包好封好口备用
- 锅中做少许油，凉油时下入白糖用铲子慢慢炒制
- 锅中的糖变成深红色时烹入酱油，下入切好的五花肉
- 不停地煸炒五花肉至糖色裹匀并微微出油
- 下入 60°C 左右的温水至刚好没过肉
- 下料酒、放入香料包后大火做开
- 盖盖改小火慢慢炖五花肉至 9 成熟
- 入盐和少许糖调味后继续焖五花肉至松软入味
- 捡去香料包，大火将汤汁收到红亮浓稠即可出锅食用

以上制作红烧肉的步骤就是一个算法。首先，以上步骤针对的是“如何烹制红烧肉”这一具体问题。其次，解决这个问题包括一系列特定的步骤，比如什么时候加水，什么时候加料酒等。根据这些步骤，大家都能做出色、香、味差别不大的红烧肉。

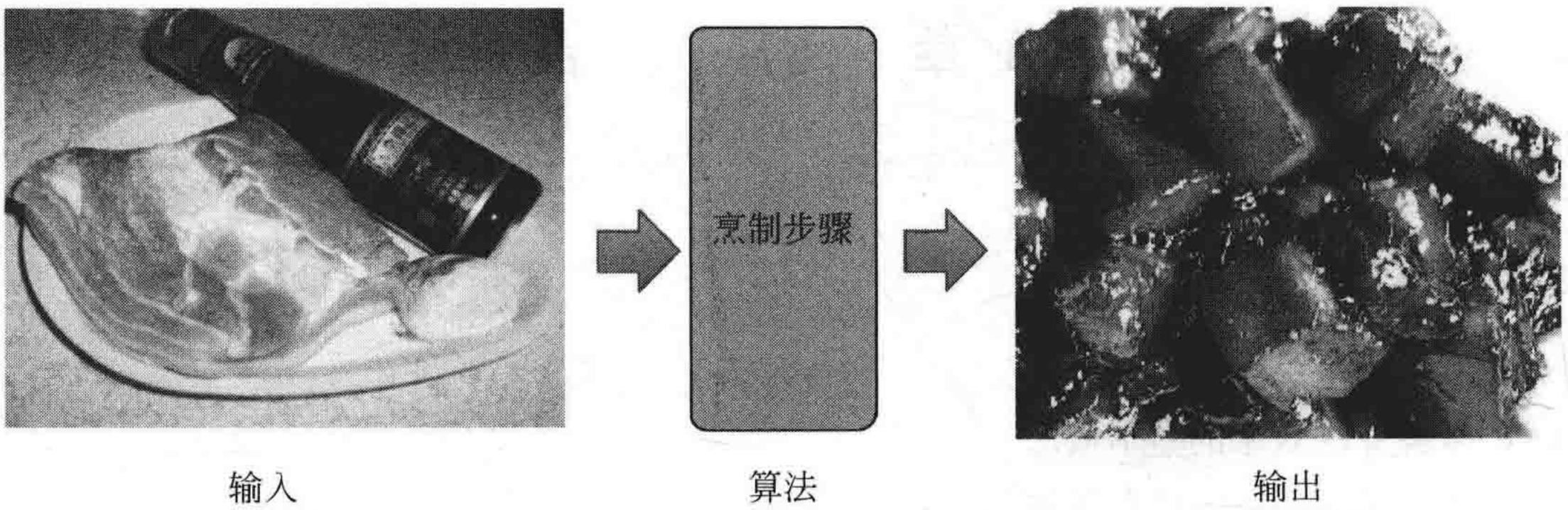


图 1.1 烹制红烧肉示意图

当然，除了以上这两个重要的属性外，我们还可以发现烹制的步骤数是有限的，且整个烹制时间大约需要一小时左右。对于一道普通家庭常常烹制的菜肴而言，一小时是可以接受的。如果一道菜的烹制时间需要一天或者是一个月，那么它就很难成为流行的家常菜肴。也就是说，一个算法除了能解决问题，还要能在一个合理的时间内得到它的解。

此外，我们描述这个“烹制红烧肉”算法采用的是自然语言。也许读者会对此有些疑问，认为算法都是用计算机程序设计的语言，如大家熟悉的 C++、Java 等来描述。其实，是否是算法和采用什么语言来描述并没有直接的关系。只要描述算法的语言能让读算法的人看懂，哪怕是自然语言也能定义一个算法。

以上例子告诉我们，算法并不神奇，我们日常的生活就会遇到各种算法。当然，算法还有其他更为严谨的定义，我们并不准备一一列举这些定义，下面将从算法的两个重要属性来进一步讨论它。

1.1.1 算法的属性

本书所讨论的是计算机领域内的算法，也就是说解决问题的类型是**计算问题**。这种情况下，算法是一个定义明确的计算过程，可以以一些值或一组值作为**输入**，产生一些值或一组值作为**输出**。因此，算法也可以说是将输入转为输出的一系列有限计算步骤。比如，前面红烧肉的例子中，输入就是五花肉和各种配料，输出当然是如图 1.1 右图所示的红烧肉。

算法的第一个重要属性就是正确，也就是说能正确地求解问题。对于一个不能得到问题正确解的算法，不管这个算法设计得多么有技巧，具有何种奇思妙想，对给定的问题而言都没有意义。比如，现在的问题是烹制红烧肉，但是给出的算法却只能烹制出糖醋里脊，显然给出的这个算法对于烹制红烧肉这一问题而言没有意义。因此，设计某个问题的算法和分析该算法的前提，就是该算法要能求解出该问题的正确解。