

计算机科学导论

—— 以Python为舟

沙行勉 编著



清华大学出版社

计算机科学导论

——以Python为舟

沙行勉 编著

清华大学出版社

内 容 简 介

《计算机科学导论——以 Python 为舟》是一本内容丰实、形式活泼,同时与计算机的最新发展密切结合的计算机入门教材。计算机包含了一切可以执行程序的计算设备。本书用深入浅出的语言讲解了计算机科学的基础知识。主要内容包括计算机学什么、神奇的 0 与 1、程序是如何执行的、学习 Python 语言与数据库知识、计算思维的核心——算法、操作系统、计算机网络与物联网、信息安全等。本书不仅让读者能够清楚完整地了解如何用计算机解决问题,而且通过 Python 程序的巧妙演绎与动手实践,让读者切实体会到计算机科学的广博与趣味,带领读者体会计算机科学的美。

《计算机科学导论——以 Python 为舟》可作为计算机科学入门课程的教科书,也可作为广大读者理解计算机科学基本知识的科普读物。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

计算机科学导论——以 Python 为舟/沙行勉编著. —北京:清华大学出版社,2014
ISBN 978-7-302-37773-3

I. ①计… II. ①沙… III. ①软件工具—程序设计 IV. ①TP311.56

中国版本图书馆 CIP 数据核字(2014)第 190234 号

责任编辑:付弘宇 薛 阳
封面设计:常雪影
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:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:21.75 字 数:539 千字

版 次:2014 年 10 月第 1 版 印 次:2014 年 10 月第 1 次印刷

印 数:1~2000

定 价:39.00 元

产品编号:061411-01

作者简介



沙行勉 (Edwin Sha), 博士生导师, 2000 年起任美国终身职正教授 (Full Professor), 中国国家千人计划 (A 类) 特聘专家, 长江学者讲座教授, 海外杰出青年学者。于 1986 年获得国立台湾大学计算机科学系学士学位, 在海军陆战队服役两年后赴美国普林斯顿大学 (Princeton University) 就读。于 1991 年和 1992 年分别获美国普林斯顿大学计算机科学系硕士学位和博士学位。1992 年起任教于美国圣母大学 (University of Notre Dame) 计算机科学与工程系, 并于 1995 年起担任该系副系主任和研究生部主任。2000 年起作为终身职正教授任教于美国得克萨斯大学达拉斯分校 (UTD) 计算机科学系。2001 年曾担任计算机科学部主任。任上海交通大学、山东大学、北京航空航天大学、湖南大学、华东师范大学等客座、兼任教授或博导。2008 年被评为海外杰出青年学者, 2010 年起任教育部长江学者讲座教授。2011 年起任中国千人计划特聘专家, 现全职任重庆大学国家特聘教授和计算机学院院长。

已在相关国际学术会议及国际核心期刊上发表英文学术论文 300 余篇, 其中包括 40 余篇 IEEE 和 ACM Transactions 期刊论文。共获各类国家级教学、科研奖项 35 项以上, 其中包括美国 Oak Ridge 大学联盟颁发的杰出青年教授奖, 美国国家科学基金颁发的杰出学术发展奖, 美国圣母大学颁发的杰出教学奖, 以及世界顶级期刊 ACM Transactions (ACM TODAES) 颁发的 2011 年最佳论文奖 (一年只选一篇最佳期刊论文) 等。以大会主席身份主持多次国际重要学术会议。沙教授在教学方面深受中美学生们喜爱, 例如, 在美国从教期间, 他在每学期由学生给老师打分的教学评鉴中都得到高分。沙行勉教授喜爱中国传统文化及儒释道哲学, 以人才培养、教学育人为其终身的兴趣及志向。

出版说明

随着国家信息化步伐的加快和高等教育规模的扩大,社会对计算机专业人才的需求不仅体现在数量的增加上,而且体现在质量要求的提高上,培养具有研究和实践能力的高层次的计算机专业人才已成为许多重点大学计算机专业教育的主要目标。目前,我国共有 16 个国家重点学科、20 个博士点一级学科、28 个博士点二级学科集中在教育部部属重点大学,这些高校在计算机教学和科研方面具有一定优势,并且大多以国际著名大学计算机教育为参照系,具有系统完善的教学课程体系、教学实验体系、教学质量保证体系和人才培养评估体系等综合体系,形成了培养一流人才的教学和科研环境。

重点大学计算机学科的教学与科研氛围是培养一流计算机人才的基础,其中专业教材的使用和建设则是这种氛围的重要组成部分,一批具有学科方向特色优势的计算机专业教材作为各重点大学的重点建设项目成果得到肯定。为了展示和发扬各重点大学在计算机专业教育上的优势,特别是专业教材建设上的优势,同时配合各重点大学的计算机学科建设和专业课程教学需要,在教育部相关教学指导委员会专家的建议和各重点大学的大力支持下,清华大学出版社规划并出版本系列教材。本系列教材的建设旨在“汇聚学科精英、引领学科建设、培育专业英才”,同时以教材示范各重点大学的优秀教学理念、教学方法、教学手段和教学内容等。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

1. 面向学科发展的前沿,适应当前社会对计算机专业高级人才的培养需求。教材内容以基本理论为基础,反映基本理论和原理的综合应用,重视实践和应用环节。

2. 反映教学需要,促进教学发展。教材要能适应多样化的教学需要,正确把握教学内容和课程体系的改革方向。在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

3. 实施精品战略,突出重点,保证质量。规划教材建设的重点依然是专业基础课和专业主干课;特别注意选择并安排了一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现重点大学

计算机专业教学内容和课程体系改革成果的教材。

4. 主张一纲多本,合理配套。专业基础课和专业主干课教材要配套,同一门课程可以有多样具有不同内容特点的教材。处理好教材统一性与多样化的关系;基本教材与辅助教材以及教学参考书的关系;文字教材与软件教材的关系,实现教材系列资源配套。

5. 依靠专家,择优落实。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

教材编委会

序

为学习计算机科学的学子们提供一本内容丰富又形式活泼的书去阅读；用我多年的深思与积累为学子们揭示计算机科学的美与真意，引导大家渐入佳境——这是我多年的心愿。在计算机领域近三十年的磨炼和进步以及在美国大学里任教多年的体会，值此以国家特聘专家身份全职回国贡献的机遇，方让我能在今年写成此书。相信各位读者在读了这本书以后，将开始对计算机科学建立正确而全面的认识，随之深入，必将功力大增！

我写这本书的目的就是希望这本书成为“计算机科学导论”的经典之作。它适合所有信息专业的学生们把它用作第一堂课的教材，了解计算机科学的核心知识之所在；也适合非信息专业的读者借助它较为完整地理解计算机科学的相关基本知识；它适合各个年龄层次的读者，把它当做一本有趣又有实料的书去阅读。我相信这本书就是有这样的趣味，并且值得玩味。

以何因缘写这本书？

(1) 全世界学者普遍认为最好的大学教授应该教最基本的大学课程。这就是为什么诺贝尔奖获得者会去教基础物理或基础化学这类课程。计算机专业亦然。计算机导论非常重要，要给学习者建立正确的概念和充足的兴趣，要让学习者在第一步就有机会对这个学科的“美”有所体会，继而为往后的计算机学习打下坚实基础。“计算机导论”是最基础的计算机学科的课程，不好教。我的目的是要通过这本书规划出：教些什么？次序为何？而且我认为，最好的方法是从对计算机科学之美的领悟来引导学习者的兴趣。这本书对计算机教学的影响很大，虽然不好写，但值得倾注心血把它写好。

(2) 用于讲授计算机导论的书的作者要对计算机科学有广泛的知识——了解各种程序语言、逻辑电路、体系结构、编译器、操作系统、算法设计、复杂度分析、计算机网络、信息安全、先进多核和分布计算系统等，进而到达“读通”的境地。并且，最好具有多年的教学经验，知道如何以生动活泼的方式来引导学生。我从1982年开始进入计算机专业读书，1988年又进入美国普林斯顿大学计算机科学博士班学习，1992年博士毕业后就一直在美国高校担任教学和研究工作，这么多年来教学评鉴总是全学院的最高分之一。近

年来全职回国任教,深感因缘能力积累已经具备,决定用中文写此书以贡献给中国的莘莘学子。

(3) Python 语言的发展。计算机导论课程应该要让学生对编程有所认识和练习。传统的计算机语言,如 C、C++、Java 等,都不适合在计算机导论课程中使用。Python 则不同,它是个可以快速上手语言,虽然它功能强大(Java、C++ 有的功能 Python 都有),但是只要学习 Python 的一些基本功能就可以使用,这使它成为学习计算机导论的利器。使用 Python 语言的好处是可以不计较程序语言在形式上的诸多细节和规则,可以专心地学习程序本身的逻辑和算法,以及探究程序执行的过程。所以,这本书的例子都是用 Python 语言来编写的。我们会深入浅出地解释一些相关的 Python 的知识,让学生们不需要有任何编程的经验就可以练习和学习编程。其附加价值是可以经由这本书来学习 Python 的使用。因为 Python 的好学易用,我的博士生们基本都在用 Python 编程,而很少用 C++ 或 Java 了。Python 语言的发展是我写这本书的增上缘吧。

(4) 因缘总是不可思议,不知何时播下的种子,有一天当时机来临时就会发芽结果。究其种子的根源实已难觅,或许这个种子是在年轻时就已经种下了吧。我和许多学子一样,年轻时辄有大志,以中国读书人自居,为中华文化的衰落而忧愁,以传承圣贤之道而自勉。后攻读计算机专业,暗自庆幸做了正确的选择。计算机学问与各类学问之间的融通与相互的印证,广博多彩的应用,再加上几十年来对我人生的体验和对哲学的些许领悟,使我感觉到计算机科学是美的(其实其他学科亦然,只要是钻研到甚微和通达之处的,也必是如此感想吧)。总之,觉得是时候来分享一些粗浅的心得给大家了,也希望带动各位读者多多思考,进而分享你们各自的领悟给众人。

这本书的特色为何?

我用“体、相、用”三方面来说明这本书的特色:“体”代表的是原理和本质;“相”代表的是其缤纷的色彩;“用”则代表了其应用。

(1) 体大。包含计算机科学的基本道理,深入浅出,直指核心。从 0 与 1 的开关、程序的执行过程、解决计算问题的思路,到各种先进计算系统的介绍,更辅以对计算机科学的多个层次与角度的“美”的探索与讨论,发自基础理论,以致广博通达,此为“体大”。

(2) 相大。具备软件编程、面向对象编程、计算机组成原理、计算机系统、数据结构、计算机算法、计算机网络、信息安全等的基本知识。一本书可以作为众多课程的导论,此为“相大”。

(3) 用大。

① 循序渐进,不知不觉中学习了方便好用的 Python 语言。

② 计算算法的基础训练,从此对计算问题的解决思路能有所依循,知道如何利用计算机解决问题。

③ 对程序设计、网络、网页、信息安全等诸多重要知识,能有基本的掌握。

④ 学习如何体会一门学问的美,结合人生哲学,深自反思,以致兴趣盎然,其乐无穷。综合这四点,此为“用大”。

书名有何含义？

书名是“计算机科学导论——以 Python 为舟”，是取其大意简略而为之。假如列出全部的意思可能会太长了：“计算机科学导论及谈它的美和相关的领悟——本书以 Python 为工具”。

“计算机”在此泛指一切利用程序而执行计算的系统。这个程序可以是随意改动的软件，也可以是已经固化而不能随意改动的固件或硬件。这类计算系统包含了汽车、家电、机器、航空航天等各种领域的智能控制系统，包含了人人都有手机系统，也包含了各式各样的计算机、多核系统、分布式系统等。

而“计算机科学”是设计和应用计算机的理论、技术和工程的总括。随着时代的进步，计算机科学的范畴也越来越宽广，包含了软硬件工程、计算机网络、物联网、信息安全、大数据等相关领域。

在此我要特别提醒学习计算机专业的同学们，“计算机”和“计算机科学”这两个词汇在中文里面常常混用而不分，但是用在外文时就要注意了，不要引起笑话。计算机是个仪器，而计算机科学是门学问。譬如有人问你：你的专业是什么，你用中文回答：我的专业是计算机（或软件）。这在中文是通的，但是英文翻译过去就不通了。你不能说“My major is Computer.”（或“My major is Software.”）这是不通的话，因为 Computer 是个仪器，而不是个专业，Computer Science 或 Computer Engineering 才是专业。所以你应该回答：“My major is Computer Science.”或者“I study Computer Science.”又如，“软件学院”的翻译不是 School of Software，而是 School of Software Engineering。假若对词汇的使用不注意，汽车学院变成了 School of Automobile，渔业学院就变成了 School of Fish（一群鱼）了，那就成笑话了。因此，学科与非学科的词汇不要混淆。

“导论”是指用较为简洁的语言来论述这一学科的基本和整体的思想，从而使读者对该学科有较为正确和系统的把握。英文是 Introduction。在这个词汇上，中文的“导论”远胜于英文中 Introduction 的内涵。“导论”这个词汇有引导的含义，而英文的 Introduction 则欠缺这个含义。这本书的目的是，除了给读者做一个概括性的、深入浅出的介绍外，也要激发读者的兴趣，引导读者做更深入的学习。

“以 Python 为舟”是“以 Python 为工具”的意思。Python 这个语言是个很好上手的语言，例如要计算一些数组（存在 X 数组中）的总和与它们的平均值，可以在几分钟内写出如下的 Python 的程序：

```
X = [10, 4, 6, 9, 12, 92, 138, 26, 98, 21, 8, 98]
sum = 0
for i in X:
    sum = sum + i
print ("总和是:", sum, "平均值是:", sum/len(X))
# 注: len(X)代表 X 数组的元素个数
```

这个程序简单易懂。用循环重复（用 for 语句）执行加法的方式把 X 数组里的元素一一累加到 sum 里。最后用 print 来输出结果。至于这个 for 循环需要循环执行几次，写程序的人不需要在程序里特别写明，而是交由 Python 语言的解释器去理解。这使得程序员的负担

减轻了很多。

试想在其他常用计算机语言里,这段程序就没有这么简洁了。例如,在 C 语言里,这段 for 循环程序会是这样的:

```
for (int i = 0; i < len(X); i++)
    sum = sum + X[i];
```

C++、Java 的写法也和 C 语言类似,要定义索引变量 i ,并用 i 的数学式 $i < \text{len}(X)$ 和 $i++$ (表示变量 i 的值在每次循环时都要加 1) 明确表述循环执行的次数才行。这和 Python 的简单易懂相比,就略逊一筹了。

本书用 Python 做工具来介绍计算机科学,介绍如何用计算机来解问题,计算思维是什么,介绍程序是如何在计算机里执行的……我们提供了很多的例子,几乎所有的例子都是用 Python 语言来完成的。学生能从这本书中学到如何写基本的 Python 程序。

然而我要在此强调的是: Python 是学习计算机导论的工具,不是目的! Python 语言的功能非常强大,它是功能齐全的面向对象语言(Object Oriented Programming Language),它甚至也包含了一些函数语言(Functional Programming Language)的功能,有许多复杂有趣的功能。在编写本书时,作者忍住诱惑,不去讲述一些复杂的 Python 语言的功能和细节。例如,下面这两种定义 5×3 的全 0 二维数组(或列表)的方式是有差别的,差别在哪里? 这些细节对计算机导论这门课的学习是没有必要的。

```
a = [[0 for j in range(3)] for i in range(5)]
b = [[0] * 3] * 5
```

再举个例子: 学生不需要懂这种例子的。

```
from functools import reduce
items = [(1,1), (2,3), (9,4)]
total = reduce(lambda a, b: (1+b[0], a[1] + b[1]), items, (1,1))
```

这类 Python 的句法太复杂了,对于学习计算机导论的人而言也没什么用。其实,再复杂的语句结构都可以用简单语句复合而成,没有必要的,我认为这只是无谓的显摆罢了。这门课不需要学习这些复杂又不必要的语句。

本书会介绍面向对象编程的基本特性,因为面向对象的概念已经成为计算机编程的常识。但是,我们不会多讲 Python 的一些面向对象语言的复杂功能,这反而会模糊焦点。“以 Python 为舟”是套用佛学的用语。利用舟船来渡过大海,目标是渡过大海,而不是研究舟船的颗颗螺钉和片片甲板,不要让舟船变成达到学习目标的障碍。在计算机导论的学习中,若以舟船渡海为例,只要能掌握舟船航行的技巧就可以了。另一方面,对初学者而言,掌握 Python 的基本技巧也是个重要的学问。所谓一通百通,学习好 Python,对大家学习其他计算机语言有极大的助益。

我和许多用过 Python 的人一样,一开始用就马上喜欢上了 Python。学习 Python 有其特殊的好处。第一,它容易上手,可以马上用来自己编程解问题。有在工业界多年使用 Java 和 C++ 语言编程的资深程序设计师和我说,用了 Python 后,突然有了一种“解脱感”。我想他的话还是有些道理的吧。第二,用 Python 写程序可以让人更多地关注在创新性地解决问题的思想本质上。我鼓励同学们提出新的想法,以较少的代价(比起其他语言)来实现想法。

第三,可以建立良好的基础来学习其他的语言。大家肯定要学习很多种的计算机语言,例如 C、C++、Java、C#、VHDL 等。学了 Python 以后,再学习其他语言就会简单得多了。

“领悟”就是感想。原来想用的标题是“谈计算机科学的美”,因为太长而不用。然而“领悟”的本意就是要说说计算机科学的美。这门学问的美有各个方面,有它应用的广泛,有它对科学与工程结合,有它解决问题方法的理论之美,种种方面,难以列举。我就举一例在此描述。计算机科学是一门独特的学问,它不仅要设计出给定问题的解决方法(称为“算法”),也要研究这个问题本身的难度(复杂度)有多大。这在其他学科是很少见的,也就是它研究“问题”的本身,而不只是设计出解决问题的方案就罢了。计算机科学的“科学”之名,大体来自于对问题本身的分析吧。

我用下面几个不同的问题为例。看看这些问题的难度(复杂度)的差异。例如有 n 个数,存在于 $x[1], x[2], \dots, x[n]$ 中, n 是个较大数,如 100 000。

问题一: 求和, $\text{Sum} = x[1] + x[2] + \dots + x[n]$;

问题二: 排序,对 $x[i]$ 的值由小到大排序;

问题三: 划分,是否存在 $x[i]$ 加起来会刚好等于总和的一半。举个小例子,例如 $x = [100900230021, 58710120012, 7, 42190100005, 10011]$, 输出的答案是 YES, 因为 $100900230021 + 7$ 是数组 x 中所有数总和的一半。

在现今的计算机科学知识中,我们了解到问题一和问题二都是可以快速解决的,问题二比问题一稍微复杂一点,但是都属于所谓“多项式时间”内可以快速解决的问题,这类问题通常是以“秒”为单位可以用计算机马上解答的问题。但是问题三就不一样了,计算机科学从理论上证明这是个非常复杂的计算问题。所以到现在为止我们人类尚未找到一个快速的“多项式时间”的解决方法。当 x 和 n 的值较大时,用现在的技术找出解答所需要的时间可能要以“世纪”长的时间单位来计算了。一个是秒,一个是世纪,其差别是巨大的。而像问题三这类的复杂计算问题实际上还有很多。问题三的最直接解法是试验所有的子集合(其实只需试验其一半个数的子集合),看子集合内的数加起来是否刚好等于总和的一半。大家知道, n 个元素所组成集合的子集合数量高达 2^n (2 的 n 次方)这么多。当 n 是 100 000 时,这个数是比天文数字还大的数字! 然而,当 $x[i]$ 的最大值不太大时,用“动态规划”的技术来解这个问题是较快且实际可行的解决方案。“动态规划”技术会在本书的第 5 章讨论。但是对于 $x[i]$ 的最大值很大时,动态规划的解也要花很长的时间了。

有趣的是,这些复杂问题的存在对于人类来说并不一定是坏事。我们的信息安全的加密技术常把这类复杂计算问题用作防御信息泄密的利器。请看以下的问题。

问题四: 因数分解,给定 Z , 这个 Z 是两个 200 位长的不同质数 X 和 Y 相乘而来的,找出 X 和 Y 的值。举个很小的例子,例如输入是 12233883694360273618474283231, 输出是 241364017659577 和 50686443708503。

这个问题是我们天天在网上交易时所用加密算法 RSA 的武器,因为我们至今没有找到能快速解这个问题的算法(已知的算法都要花几百年的时间才能算出 X 和 Y 来)。正因如此,黑客就没有办法快速分解 Z , 而找出我们的密码 X 和 Y 的秘密。也可以说,现在我们全球的网上金融交易、购物之类,人类社会的重要经济命脉竟然就是建筑在这个看似非常简单的问题四(因数分解)的复杂度上。怎不令人惊叹啊! 假如某位读者有一天能找出快速求解的算法来,其影响力会是惊天动地的。假如这个人秘而不宣其快速求解的方法,因为对全球

金融影响太大了,我想此人的安全也会受到威胁吧。

从对问题四的因数分解问题的理解中延伸出来,我们的人生不也是有许多的相似之处吗?第一,一旦相乘难分解。这两个质数是“因”,乘积后是“果”。种因得果,看似简单,但是一旦得果,想要返回就困难了。所以一般人总是患得患失,生怕结果不尽人意。这就是苦恼的来源之一。然而大智慧的人是注重“因”,而不注重“果”,所谓“菩萨畏因,众生畏果”。因为果从因来,只要尽力去做“因”就是了,至于结果是什么就不必在乎了。如我们考试,重点在准备,只要尽力去准备,就无愧于心了,而不是抱着侥幸心理来得到结果。归根结底的问题是我们问自己是不是有尽力准备了呢?

第二,复杂的问题不是坏事。同样,烦恼也不是坏事,而是可以把烦恼转化成为智慧。烦恼是智慧的种子,此话一点不假。从火中生出的莲花,才是最美的莲花。不要惧怕烦恼,而是要转化烦恼,让烦恼成为获得智慧的正面动力。聪明的你,请多想想。

各章及其功用为何?

第1章:计算机学什么? 描述计算机的广大的应用,以激兴趣。讨论“计算机”是什么,以正其名。谈过去、现在、未来,以知往来。接着简述计算机系统、硬件、软件,以知其廓,辅以用 Python 实现的求解平方根的几种不同的程序,以表算法之美。本书最大的特点就是没有什么虚话,第1章就直接利用实际的例子,指出计算机科学的核心。写出平方根解的程序并不简单,本章利用 Python 写出三个程序,第一个程序简单但是性能不高,第二个程序利用二分法技术,效率提高不少又学习了基本算法技术,第三种方法最迅速,利用函数微分求切线的基本数学,可以几步就算出精确的平方根。一步步的优化,尽显计算机科学的美。本章也将简述现在前沿的应用之一,那就是大数据的应用,用许多例子来讲述数据分析对我们社会的益处。我们也会谈论用大量数据的方式来计算圆周率的做法,和对数据分析与逻辑推理的正确态度。最后讨论计算机科学的美,我们从应用和知识面的广阔这两方面来讨论。

第2章:神奇的0与1。 本章介绍二进制和其他进制的转换及其原理。组成计算机的计算能力的基本元素是二进制0与1的开关。这些开关可以由0或1的控制信号来决定开或关的输出状态,开与关的输出状态又分别成为0或1。所以,输入的控制信号与开关的输出状态都可以用0或1表示。例如,输入控制信号是1的时候,开关状态是0;输入信号是0时,开关状态是1。开关的输出又可以变成其他开关的输入控制信号,而这些许许多多简单的二进制开关就能构建出任何的逻辑运算。本章会向读者显示逻辑的威力。逻辑可以实现加法运算,加法竟然是逻辑做出来的,这让一般同学较难理解,大家可以想想自己小时候是怎样学会加法的,可能大部分人是用数数法吧。但是计算机的加法器是用许多开关构建而成的,加法这个最基本的运算对于计算机而言是一系列逻辑运算的集合,这确实有趣!而在有了加法和负数的二进制表达方式后,就可以做减法了。然后,乘法也可以实现了。其实整个计算处理和控制单元都是用这些开关组成,用逻辑运算实现的,令人叹为观止。用二进制单元还可以构建出存储单元和图像单元,把存储、计算处理,以及输入/输出单元综合起来可以构建出无比复杂的计算系统。本章最后谈0与1的美。这一章是将来学数字电路、计算机组成原理、体系结构等课程的基础。

笑话一则:沙老师问小明,有一个东西,它开了(Open)就暗了,它关了(Close)就亮了。请问是什么东西?

答案是：电灯。因为电路 Open 是断开，是断路。电路 Close 是闭合，是通路。开灯的正确讲法要说 turn on the light 或 switch on the light，而不是 open the light。自然语言是很有趣的，有时候是模棱两可的，有点恼人。

第 3 章：程序是如何执行的？任何的计算机都包含了中央处理单元(CPU)和主存(Main Memory)。中央处理单元是负责计算或读取数据的，而主存是负责储存程序和数据的。这一章讲述 CPU 是如何从主存中读取一行行的程序指令(机器语言)，所读的程序指令会指使 CPU 做计算或者去读写主存上的数据。本章将会清楚地描述计算机程序的执行过程，CPU 和主存的关系和互动，使读者对计算机程序的工作方式有正确的知识。另外，函数调用是程序执行中最重要的知识之一，本章描述 Python 是如何调用函数的，接着描述执行函数调用时对于变量和返回地址的管理。本章也会描述几种最常用的程序语言，例如 C、C++ 和 Java。本章最后谈计算机程序的美。这一章是将来学习计算机编程语言、计算机组成原理、编译原理和操作系统的基础。

第 4 章：学习 Python 语言。本章列出 Python 语言的最重要的知识。同学们只要知道这些基本的 Python 知识就可以编写程序了。当学习一个新的程序语言时，最先要了解的是些常用的内置数据结构。例如，Python 的列表(List)和字典(Dictionary)是很强大的。要用 Python 写程序的同学一定要熟悉列表和字典的使用。Python 字典是符合数据库数据表格的概念。接着讲述 Python 的自定义数据结构——类(Class)和面向对象编程的基本概念，我们会用数据库应用的一个例子来阐释面向对象编程和数据库的基本概念。在此例中，学生们组成一类数据，课程们也组成一类数据，学生和课程间彼此建立了关系：每一个学生具备所修习的课程名称，每一个课程具备了所修习学生们的学号信息。这个例子讲述如何基于面向对象来建立数据类组，如何建立关系，如何利用关系来完成学生选课、考试和计算平均分数等功能。最后，我们讲述如何用 Python 自带的有趣的小乌龟(Turtle)来画图，可以画出很多复杂的图形来。我们会画出一个迷宫来，读者在下一章“老鼠走迷宫”时可以用到。

这一章是很独特的，坊间还没有任何一本 Python 书能达到我们精简求实的目标，在短短的一章中全面讲述 Python 语言的各种性能，最有用的是我们的“经验谈”，读者请多体会和学习。学习任何语言(包括计算机语言)的诀窍是“多看多写多玩”，要多“玩弄”Python 就是了。我相信 Python 比任何游戏都好玩。这一章是学习任何其他程序语言的基础。关于数据库方面，本章也利用 Python 语言独特的字典数据结构来介绍数据库的基本知识，另外，本章以面向对象的编程方式来实现课程和学生的数据库基本功能。

Python 也有烦人的地方，Python 语言第三版(v. 3.0)以上和第二版的有些用法不兼容。例如，在第三版及以后的版本中，print 后面一定要有括号，而 raw_input()不能用了。请注意，本书使用的是 Python 第三版以上的版本。所以大家在看 Python 教程时，要选择第三版以上的教程。另外，Python 语言和所有语言一样，总有些不统一的规则，大家需要小心，这些确实是比较烦人的。而我们这本书是注重计算的基础知识，所以不会太着重于这些特例的学习。

例如，下例显示出一个语言的实现细节会导致程序输出结果与数学结果不一致。

```
>>> x1 = 123456789
>>> x2 = 2097657821235948841
>>> y = 19
```



```
>>> z1 = x1 * y
>>> z2 = x2 * y
>>> int(z1/y)           # int(x)代表是取 x 为整数的值
123456789              # 正确, 等于 x1

>>> int(z2/y)
```

2097657821235948800 # 竟然不等于 x_2 , 请问要如何写使得 $z_2/y == x_2$? (答案见第 4 章)

再强调一次, 本书是用的 Python 第三版以上的版本。

第 5 章: 计算思维的核心——算法。这一章非常重要, 也是本书的亮点之一。计算机科学的美尽在此显现。一个大问题的解决方案是由分立的小问题的解构建而成的。具体而言, 就是对递归概念的扎实学习。希望各位同学从现在开始都尽量用递归的方式思维。而这种思维方式是大部分同学在接触计算机算法之前很少学习的方式。例如在一个平面上, 一条线可以分出两个子平面来, 二条线可以分出 4 个子平面来, 那么 n 条线最多可以分出多少个子平面来? 假如 $F(n)$ 是 n 条线最多分出的子平面个数, 只要我们找出 $F(n)$ 和 $F(n-1)$ 的关系, 这个问题就马上能解出来了, 大家试试看。解答会在第 5 章。

在递归的概念下, 我们学习分治法和动态规划。这两个方法是解问题最常用的方法。而我们在日常生活中所最常用的贪心算法是有其局限性的: 贪心算法很少能得出最优解来, 不可不知啊。例如, 从 A 地开车到 B 地, 要经过多条道路, 要如何选择最快速的路径? 贪心算法是在 A 处选择最通畅的那条路。然而, 这条道路之后可能就是堵塞的道路。贪心算法只顾眼前, 而不看全局, 通常没有能力找到最优的全局解。本章用“老鼠走迷宫”为例, 看我们如何用计算思维来轻松地解决问题。学习了计算思维后, 对我们的人生做人处事也会有所影响吧。这一章是同学将来学习数据结构、算法分析与设计的基础。

第 6 章: 操作系统简介。操作系统是计算机教育中最重要的知识之一(另外还有算法和体系结构), 学生一定要牢固地掌握操作系统的知识。不管是手机(安卓、IOS)还是电脑(Linux、各类 Windows)都是以操作系统为硬件接口和软件平台。本章首先会解释开机时系统内部经过了哪些步骤, 接着讲述操作系统最重要的概念——那就是操作系统是世界上最懒惰的东西。它的正常状态是睡觉, 它只有被“中断”吵醒后才会做事, 做完事后再睡觉。对“中断”的深刻理解是至关重要的。

一个计算机系统内会有多个程序(可能几十个程序)在同时执行, 然而在硬件上, 只有相对较少的 CPU 核可以使用, 这就需要操作系统来管理程序和所有硬件资源。操作系统的作用有:

- ① 管理各种外围硬件设备, 例如 U 盘、网卡、键盘等, 并管理文件系统;
- ② 管理程序共享的资源, 例如 CPU、主存等(一个计算系统会有许多程序同时在执行或等待执行);
- ③ 管理和调度多个程序的执行;
- ④ 提供程序和硬件的衔接, 即提供各种系统的服务和接口。同学们一定要理解操作系统的内部机制, 这样将来才有能力实现出一个“计算机系统”。本章也会讲述“文件系统”, 以及 Python 是如何读写文件的。这一章是将来学习嵌入式系统、网络、信息安全的基础。

第 7 章: 计算机网络与物联网。计算机网络是个非常复杂的系统, 当我们在 QQ 上发

送一个笑脸给对方,这个笑脸的传递是经过层层转换和编码,经过环环的连接和控制,其复杂的程度可以说是人类科技文明的高度展现。这一章简洁地介绍各个网络的层面,学完这一章将会对计算机网络有一个正确而全面的认识。另外,这一章介绍网页的原理,网页访问的流程,静态和动态网页的差别,举出网页制作的实例。这一章将是学习计算机网络课程和网页制作课程的基础。

互联网的触角将不断延伸,逐渐渗透到我们的日常生活中,从而催生一种新型的网络——物联网(Internet of Things)。物联网被认为是以物品为载体,通过射频识别技术等传感设备与互联网建立连接,从而实现物与物之间的互连。在物联网的时代,每一个物体都可以寻址,每一个物体都能实现通信,每一个物体都能控制。这样的物联网时代将让我们充满期待。这一章所介绍的物联网将是物联网相关专业课程的基础。

近年来计算机网络的发展给人类文明与商业模式带来巨大的改变。通常我们身处在这个改变中而不自知。我们理所当然地习惯了这些改变,信息在网上得知,购物在网上交易,娱乐在网上享受,朋友在网上结交……是方便了,但是当我们把情感的交流诉诸于网络,人与人之间的形式交流好像是快了,但是深层的感觉又好像远了。交流中多了即时反应的只字片语,而少了静下心来沉淀积累。我想好多人没有在深夜花几个小时写封沉甸甸的信给远方的亲人了吧?

第8章:信息安全。信息安全这门课是我最喜欢教授的课程之一。它的内容有趣,一方在进攻,另一方在防守。要全面地了解各种黑客进攻的手段,才能较完整地学习各类防御的技术。首先谈计算机的常见威胁,包含了客户端的威胁、服务器端的威胁和网络上的威胁。接着讨论各类安全技术。对信息安全而言最重要的是密码学,因为密码学是很多防御方法的基本技术。例如A要送一个信息M(例如信用卡密码)给B。我们怎么知道信息M在中途有没有被人改动过?这就需要密码学的技术。B收到M后怎么知道这个M是从A来的?这就需要密码学的技术。我们怎么能保证M的原文在途中没有人能偷看?这也需要密码学的技术。

入侵检测、防火墙、网络安全、杀毒软件、系统安全都会在这一章讨论。新型的手机系统安全也会涉猎。最后谈谈对信息安全的领悟。这一章和前面各章的集合都是信息安全专业的基础。

这本书要如何作为教材?

这本书是以计算机导论课程的教材为目标而写的。适用于32~48学时的课程,宁慢勿快。不要只是单方面传授,不要填鸭,要制造活泼的课堂气氛,引导学生学习,多在课堂上讨论,多问学生问题。相信老师们在教这门课时会对计算机科学产生更多的心得和体会吧。

本人自1992年在美国高校任教以来,以教学为乐。常年来在美国高校课堂上被学生们评价为教学最好的老师之一。我在课堂上与学生的互动很多,会准备很多有趣又重要的问题来问学生,效果很好。也给学生较多的作业和有趣的需要动手的课程设计项目,不怕他们开夜车,花时间完成作业。老师在关怀学生的同时需要严格要求。学生们是聪明的,他们知道这个老师是花时间和精力来教导他们,他们会感激这样的老师的。中国的莘莘学子们常年来受应试教育的摧残,尤其是一些大一的学生们,可能还是以应试为目的,老师不划重点就不知道重点,没有大考小考就没有学习的动力,所以我的建议如下:

(1) 要让学生们知道这门课是他们学习计算机的最重要的一门课！只有多花功夫才行。

(2) 多些随堂小考试。这些随堂考试可以达到点名的目的，也可以督促他们的课后阅读与学习。

(3) 所有书里的程序都要求学生去试验，要求学生去改进，要求学生去“玩”编程。

(4) 上课要有趣味。不要“教死书”。要旁征博引，多互动。上起课来收放自如，先提核心，引起疑情，不讲答案；如侦探小说一般，埋设疑点，再铺陈开来，以激发寻求解答的好奇心；条理分明，例子多些；活泼气氛，互动多些。到了讲台上就要潇洒点，这潇洒是来自于自己的学识、素养和充分的准备。

(5) 及早和定期给学生布置作业。学期刚开始要多布置点作业，让学生养成好习惯，知道这门课的压力大，他们才会预留多点时间给这门课。作业分两种，一种是较理论型的作业；另一种是要动手做的作业。可以分别布置。建议要早点给学生布置作业，第一个星期就布置作业也不嫌早。学期一开始就要给学生正确的学习方法和要求。所谓一鼓作气，再而衰，三而竭。

计算机科学导论课程是计算机课程体系的第一门课，至为重要。最好有实验课配合教学。作者所在大学的计算机导论实验课有 32 学时，包含了 16 学时的 Python 编程和 16 学时的机器人编程（也是用 Python 语言）。学院购置了足够的机器人让学生们编程，以激发他们的兴趣，收效不错。

这本书的前 6 章最为基础，需要花较多的时间。假如没有时间，后两章可以简略教授。可能的课程计划如下：第 1 章：2~4 学时，第 2 章：6~8 学时，第 3 章：4~6 学时，第 4 章：6~8 学时（若有实验课，可在实验课教学，总之有机会就让学生多动手），第 5 章：8~10 学时，第 6 章：4~6 学时，第 7 章：6~8 学时，第 8 章：6~8 学时。让学生清楚掌握前六章的内容，这门课的教学就算是成功了！

这本书里还有三位代表性人物：“小明”是位活泼好问的学生，“阿珍”是位认真负责的研究生助教，而“沙老师”代表作者和授课老师。他们在本书中的问答，还是有些深意的。

为了让老师能较为方便地讲授，我们准备了所有章节的 PPT 课件供老师们使用，也提供了书中所有实例的源代码和习题答案供读者们执行和修改。这些配套资料请从清华大学出版社网站 www.tup.com.cn 下载，下载与使用中的相关问题请联系 fuhy@tup.tsinghua.edu.cn。

与本书相关的辅助教材请见我的网页以得到最新信息：

<http://blog.sina.com.cn/edwinsha>

<http://www.cs.cqu.edu.cn/public/tindex/B0226>

致谢

要感谢的人很多。首先感谢我的家人——我的妻子诸葛晴凤教授和我的女儿沙奕兰，是她们的支持和鼓励，使我可以花时间和精力来完成这本书。为了这本书，我时常工作到深夜，谢谢家人的体谅和我妻子对撰写这本书的实质性帮助。除了家人外，还有就是重庆大学我的研究团队，他们在汲汲于科研之时，还要帮助我来撰写这本书，我很是感激，主要领头人

有谷守珍、姜炜文、陈咸彰等,还有很多都是做了特殊贡献而在此没有列出名字的,因为一定会有疏漏,还是不列为好,然而点点滴滴,记在心头,我一并感谢。我们心中抱持的信念是一致的,那就是写出一本最好的计算机导论的书,对中国计算机的教育做实质的贡献!

整本书的内容和例子都是基于我的想法组织而成的,没有参考或拷贝世界上任何一本计算机科学导论的书。大部分的 Python 程序都是我自己编程和试验的。这本书是我们花心血写成的。然而,疏漏之处,在所难免,我们要持续改进,所谓任重而道远,此之谓矣。我们已经在计划将来的新版和辅助内容了,肯定会越来越好的。

最后祝福大家天天快乐。

作者

2014年8月