

A sunset over the ocean with the sun low on the horizon, casting a golden glow across the sky and water. The sky is filled with soft, orange and yellow clouds. The water is dark with a shimmering path of light reflecting the sun.

计算机问题求解—论题1-1
- 为什么计算机能解题

2022年09月19日

关于本课程能力目标的说明

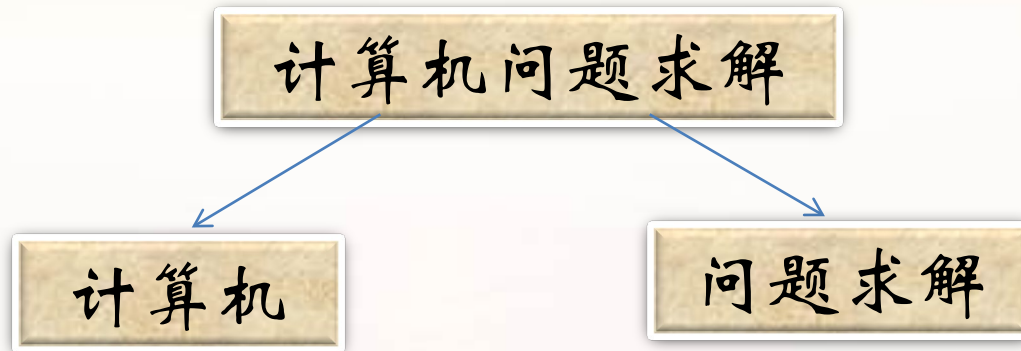
- 具备分析问题，采用一定策略进行算法设计的能力；并具备对算法进行基本分析的能力；
- 具备自我探索学习，并凝练问题的能力；

你认为为了应付快速变化的社会与技术环境，
什么是最关键的个人能力？

你认为“学习能力”的核心是什么？

你理解什么是 **Critical Thinking** 吗？

本课程的基本问题



问题0a:
计算机究竟能干什么？

问题0b:
人究竟是如何解题的？

问题0：为什么计算机能帮我们解题？

- 计算机究竟能帮我们解什么样的问题？
- 如何能让计算机更好地帮我们解题？

Part I

了解计算机

- 以“问题求解”的观点

Amazing Machine

Computers are amazing machines. They seem to be able to do anything.

- 承担社会生活中的各种各样的任务
- 具有令人惊叹的精确性
- 使人从巨大的事务性负担中解脱出来
- 为人提供娱乐，同时也能为人创造出更好的计算机
- ...

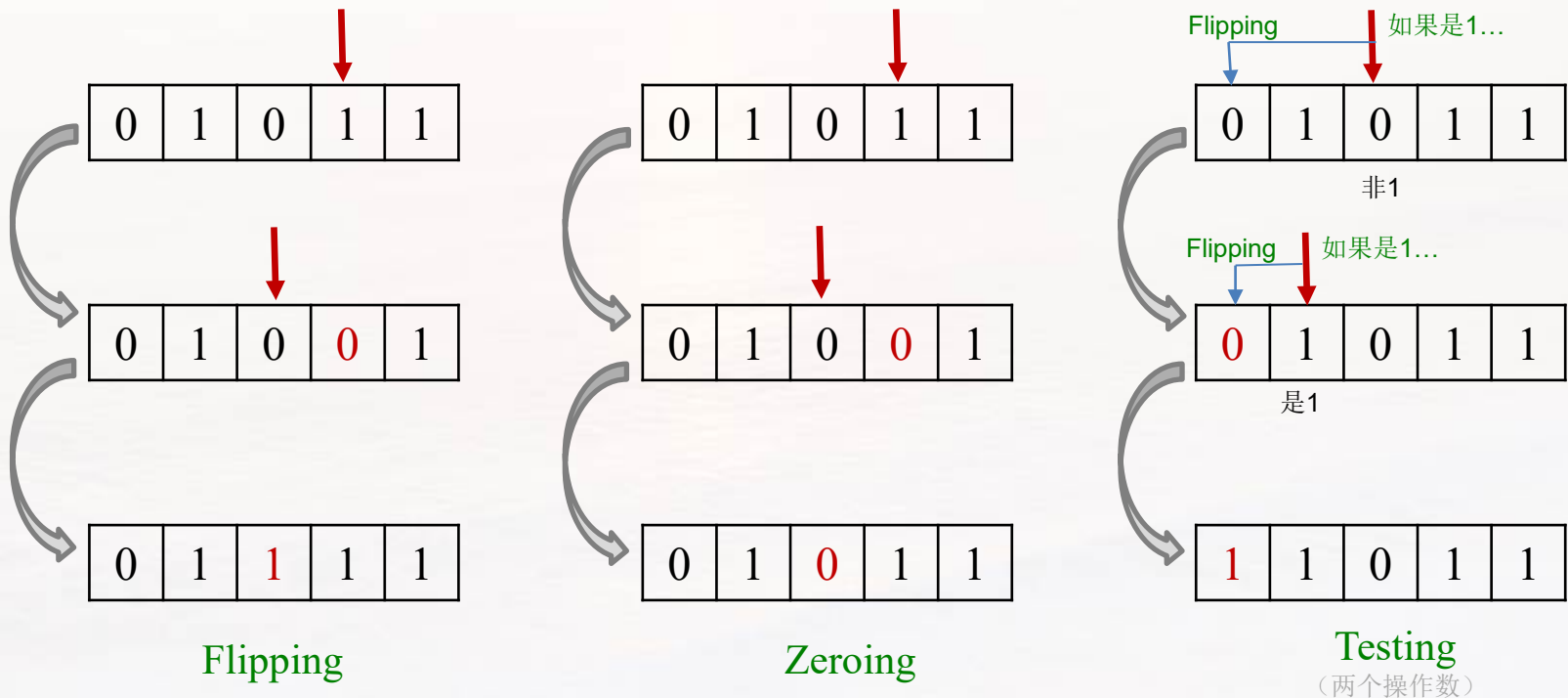
问题1:

如果从解题的角度让你“极度抽象”，你会如何想象计算机这个“黑匣子”的内部结构呢？

Even More Amazing

A computer can **directly** execute only a small number of **extremely trivial** operation.

- 本质上说，一步改变一个cell的内容



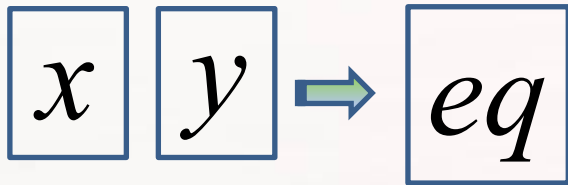
注意：每个operation有相应的operand (1个或多个，也可能是0个)

问题2:

既然“直接”干不了什么，那么“间接”呢？

问题:

判断任意给定的两个一位二进制数(bit)是否相等。



If $x=y$ $eq=1$
Otherwise $eq=0$

Equality test (x,y)

```
zero eq;  
flip eq; /* equality on  
test x flip eq;  
test y flip eq;
```

你能将这个操作扩展到，比如，32位内的整数吗？

问题3:

为什么这个结果肯定正确？

Thinking as Computation
还是
Computation as Thinking

增加两个操作，可间接地利用判断相等的操作 goto, exit 实现加法

计算两个1-bit二进制数的和

add(x,y)

1. zero z0;
2. zero z1;
3. **equality test**(x,y);
4. test eq **goto** 7
5. flip z0; /*x,y不同，和为1(01)
6. **exit**;
7. test x flip z1; /*x,y是1，则结果是2(10)

x

y

x+y

z1	z0
----	----

问题4:

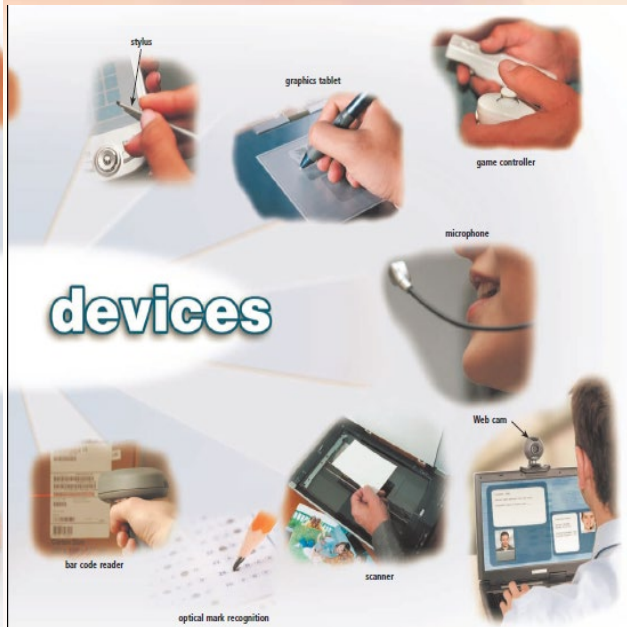
你能说出这新增操作的关键价值吗?

如果只允许原来的三个操作，你还能解吗？困难在哪里？

多层次抽象

- 用一位的加法“间接操作”可以实现普通加法操作；
- 加法操作又可以作为一步操作用在更复杂的“间接操作”中。
- 实际上现在计算机内部电路能提供的操作远不只是那几个最简单的“直接”操作。但也仍然是“很有限的”数量。

但我们“看到的”计算机可不是只能处理bits的？



input devices

输入
Input

输出
Output



内部与外部

However, the outward appearance is **of peripheral importance when compared to the bits and their internal arrangement**. It is the bits that “sense” the external stimuli arriving from the outside world via buttons, levers, keys on a keyboard, electronic communication lines, and even microphones and cameras. It is the bits that “decide” how to react to these stimuli and respond accordingly by directing other stimuli to the outside via displays, screens, printers, loudspeakers, beepers, levers, and cranks.

问题5:

区分“内部与外部”对于让计算机“什么都能干有什么意义？

这其实还是“抽象”的力量

问题6:

现在你能回答“计算机究竟能做什么”了吗？

也许可以这样回答：

计算机本身做不了什么，但在人的“指挥”下，计算机似乎什么都能做，因为“间接”有无限的“想象空间”。

那人如何“指挥”计算机呢？

可能你立即会想到：“编程序”？那么什么是“会编程”呢？

Part II

了解 Problem-Solving

- 以“利用计算机”的角度

我们如何解题？

- George Polya: “How to Solve It?”
 - **Understanding the problem**: “What you are given and what you are supposed to figure out”
 - **Devising a plan**: “How will you attack the problem?”
 - **Carrying out the plan**: Solve the problem.
 - **Looking back**: check the result, and...

问题7:

计算机能帮我们做什么？

我们如何解题，用计算机？

- 计算机如何理解问题？
 - 输入是什么？输出是什么？
- 如何针对计算机制定计划？
 - 什么样的“计划”可能在计算机上实现？
 - 什么样的形式才能让计算机知道该怎么做？
- 执行计划 – “计算机解题”
 - 只有这个才真正是计算机做的！
- 回头看
 - 为什么结果是正确的？
 - 效率能提高吗？

问题8:

怎样才能让计算机
帮到我们?

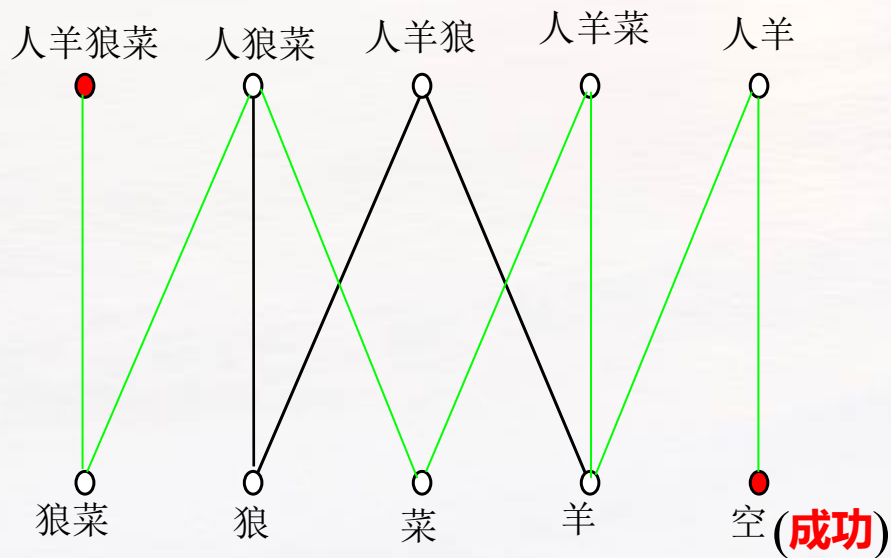
一个例子 - “渡河问题”

问题：人、狼、羊、菜用一条只能同时载两位的小船渡河，“狼羊”、“羊菜”不能在无人在场时共处，当然只有人能驾船。

图模型：顶点表示“原岸的状态”，两点之间有边当且仅当一次合理的渡河“操作”能够实现该状态的转变。

起始状态是“人狼羊菜”，结束状态是“空”。“允许状态”只有10个。

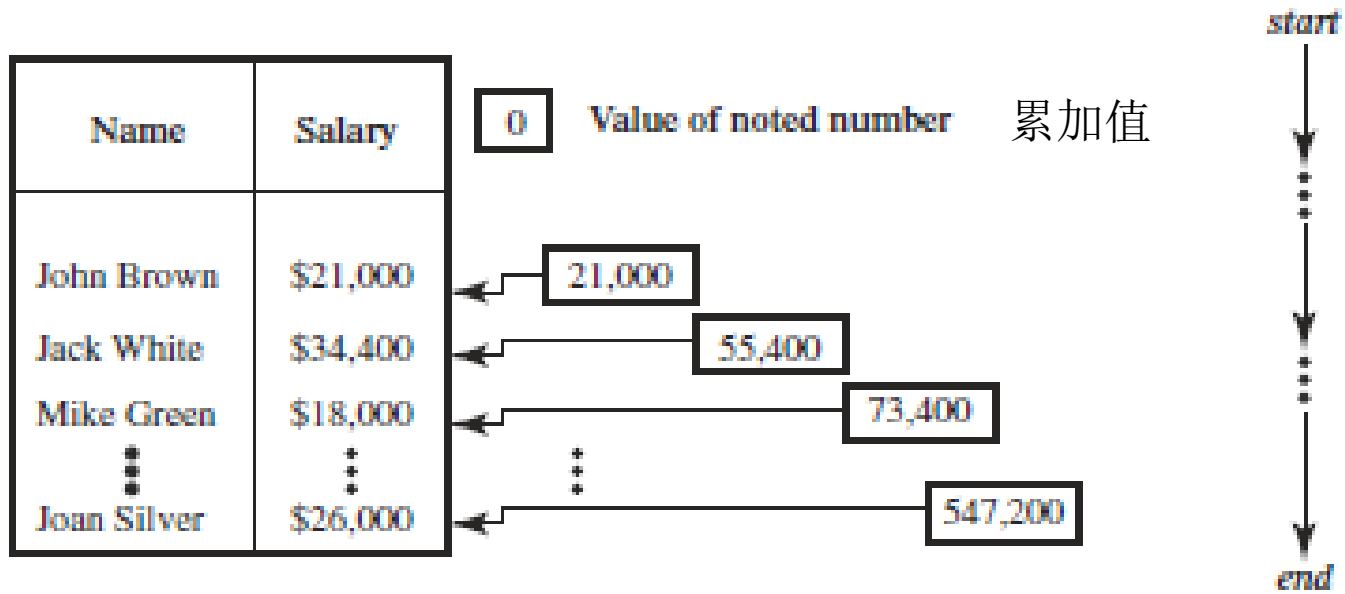
问题的解：找到一条从起始状态到结束状态的尽可能短的通路。



计算机问题求解与数学

- 对问题的理解必须用严格的数学语言描述。
- 其前提是必须建立问题的数学模型。
- 可用的数学模型必须是计算机能对其进行操作的。
- 让计算机能理解的解题plan必须建立在严密的数学基础上。
- 将plan表示为计算机能执行的“指示”的语言必须建立在严密的数学基础上。
- 分析计算机计算的结果必须使用数学方法：
 - 用逻辑证明结果正确；
 - 动用必要的数学手段分析解法的效率。

为什么简短的指示能让计算机做大量计算？



问题9:

还记得前面问你如何将1位计算扩展成32位计算吗，与这里有何不同？

题解 = “算法+数据结构”

问题10:

你知道Google(或者百度)
“背后”有什么吗?

它为什么能找出你要的内容, 而且很快?
它为什么能将你可能最想要的放在最前面?

问题11:

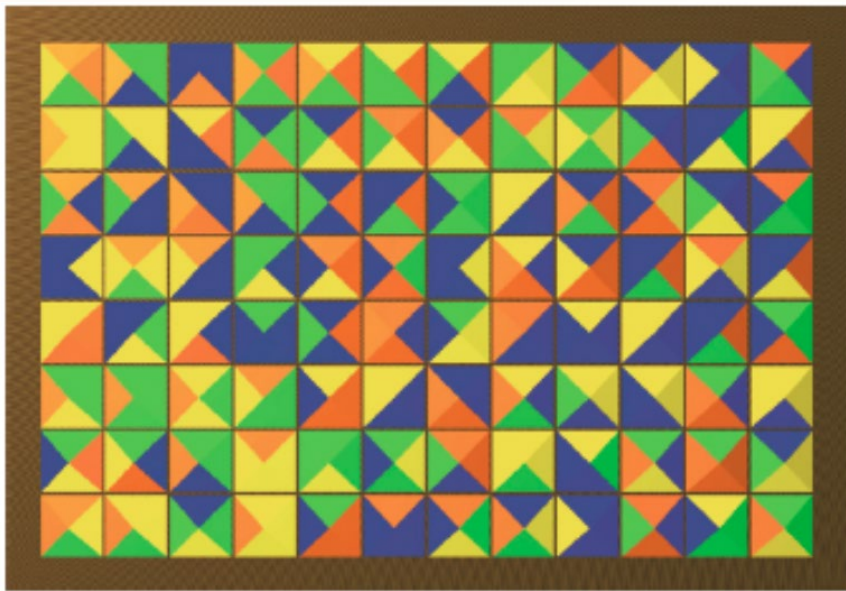
计算机下象棋、围棋能战胜世界冠军, 识别人的能力还比不上三岁小儿, 为什么?

“人不知道怎么解”, 与“知道怎么解却解不出来”

问题12:

你对计算机的“智能”是如何理解的？又是如何期待的呢？

“智能” 背后：方格拼图游戏



问题描述：

8×12个染色方块构成矩形结构。给定任意格局后每个方块不可以旋转；可选择任意两个方块互换位置，此操作可执行任意有限多次。

计分：边界相邻的两侧颜色相同得1分，否则0分。

游戏目标：尽可能得到高分。

理解问题：分解：交换方块导致分值变化，计算分值确定是否有收益，一般情况下似乎无法判定当前是否最高分

结构：方块及其着色；置换及导致的分值变化；计分对象

建模：方块及着色；分值结构与升降；分值相关边

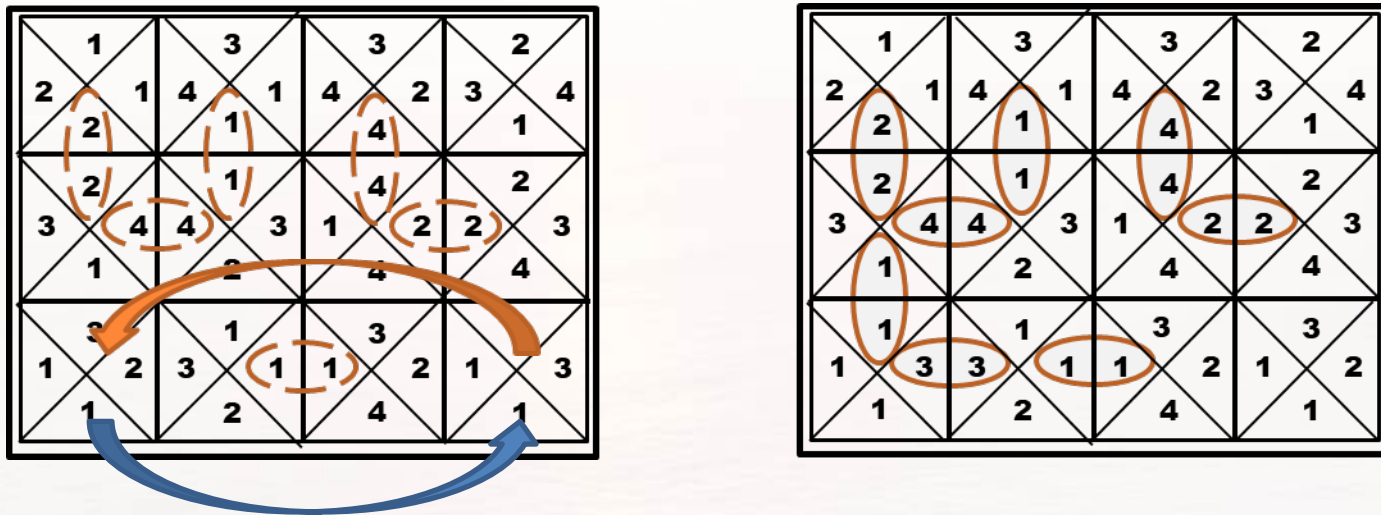
算法：似乎没有什么办法判定任意的置换是否对全局有正或复面影响

基本模式：

方块：每个方块格局一样，只是颜色布局不同

计分模式：遍历每条（内）边，但有上/下和左/右两种模式

互换：两个被置换的方块导致分值变化的计算是一样的



左边置换操作导致
总分值加2

对 $m \times n$ 输入格局建模：

每个方块可以用集合 $\{1, 2, 3, 4\}$ 上的四元组表示

将整个拼图矩阵表示成小方块的数组，一维与二维都可以，计算分值并不需要显示表示每条边，只需用数组下标控制

关于算法的考虑：

- 穷举法显然不现实！
- 但实在想不出有什么有效的策略，好在计算机计算很快，那就“碰运气”吧

碰运气也可以变变花样

铺砖算法

多次反复执行：

- 1 温度降低少许（其实只是个“抽象的”温度值）
- 2 随机选择要交换的两块花砖
- 3 如果交换后游戏得分值下降：
 - 4 随机决定是否保留这次交换
 - 5 保留交换的概率值随着温度值下降逐步下降
 - 6 如果决定不保留则撤销本次交换操作

你觉得从不容忍得分下降会是更好的策略吗？

现在你能试试看编个程序实验一下吗？

当然编程必然有很多细节需要考虑！

做一点广告

方块游戏程序如何实现以及部分运行实验结果可见下面这本书



本书主要特点

01

严谨性

南京大学计算机科学与技术系教授陈道蓄与北京大学计算机科学技术系教授李晓明合著，凝聚多年教学与科研经验。

02

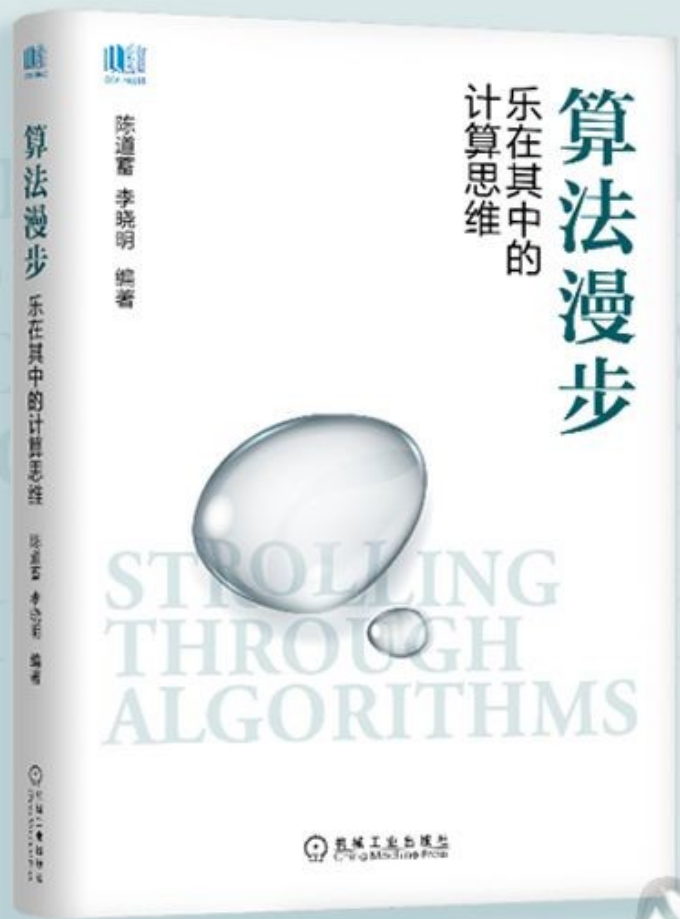
通俗性

从益智游戏、趣味故事、生活案例、经典算法等角度出发，引发兴趣，逐步进入算法世界。回避严格的数学推导，让阅读更加清晰、畅快。

03

可读性

多采用伪代码及Python形式代码讲解算法，方便学习不同计算机语言的读者进行阅读和理解。



计算机“力量”的核心：

包括“智能”的来源

- 算力：可能你没进过“大机房”，那你知道买笔记本电脑时，CPU的指标是什么意思吗？
- 算法：你知道什么“例子”能说明算法对我们生活的巨大影响吗？
- 数据：你知道为什么数据前面加上一个“大”字，就变成“宝贵资源”了？

计算环境的革命为你 - 不管你是从事那个领域的工作 - 提供了无限的创新机遇，就靠你的计算思维能力来发现了。

算法是用计算机解题的关键

至少在可以看见的未来仍会如此

Algorithmics is more than a branch of computer science. It is the **core of computer science**, and, in all fairness, can be said to be relevant to most of science, business, and technology. The very nature of algorithmics renders it particularly applicable to those disciplines that benefit from the use of computers, and these are fast becoming an overwhelming majority.

问题13:

阅读材料中紧接着上面一段话后面有关于“计算机科学”这个名词的讨论，你看了吗？有想法吗？

As soon as an Analytical Engine exists, it will necessarily guide the future course of the science. Whenever any result is sought by its aid, the question will then arise – By what course of calculation can these results be arrived at by the machine in the shortest time?

- Charles Babbage, 1864

课后作业

- UD 1.2 – 1.6; 1.8
- 探讨用我们所介绍的3个基本操作实现加法的可能性，如果你认为不可能请给出你的论据。
- 采用类似渡河问题的方法解如下问题：
 - 3个容积分别为8，5，3升的油桶，没有容量刻度，最大的桶装满油，其它两个是空的，你可以将任一桶中的油的全部或部分倒入另外的桶，但不可溢出。要分出4升油，至少要倒多少次？
- 【可选】A, B, C三人统计最近100天内每人每天收到的邮件数量。如果A和B，B和C，A和C收到邮件数相同的天数分别为 x, y, z 。设计一个算法，对任意输入的三个整数 $x, y, z \in \{0, 1, \dots, 100\}$ ，输出三人收到邮件都相同的天数的范围，或者指出输入的三个数“不相容”，即不可能发生。