



COMPUTER SCIENCE *Unplugged*

Understanding computing
through games and puzzles

Tim Bell

Andrea Arpaci-Dusseau 著

Ian Witten

Mike Fellows

Isaac Freeman

Matt Powell

插画

孙俊峰 杨帆

译

电 不插 的



计算机科学

玩智力游戏 学信息知识



华中科技大学出版社

<http://www.hustp.com>



COMPUTER SCIENCE *Unplugged*

Understanding computing
through games and puzzles

Tim Bell

Andrea Arpaci-Dusseau 著

Ian Witten

Mike Fellows

Isaac Freeman

Matt Powell

插画

孙俊峰 杨帆

译

电 不插的 计算机科学

玩智力游戏 学信息知识



NLIC 2970653479



华中科技大学出版社

<http://www.hustp.com>

中国·武汉

图书在版编目(CIP)数据

不插电的计算机科学/Tim Bell 等著;孙俊峰 杨 帆 译. —武汉:华中科技大学出版社, 2010.11

ISBN 978-7-5609-6062-3

I. 不… II. ①T… ②孙… ③杨… III. 信息技术-青少年读物 IV. G202-49

中国版本图书馆 CIP 数据核字(2010)第 046090 号

不插电的计算机科学

Tim Bell 等著 孙俊峰 杨 帆 译

策划编辑:徐晓琦

责任编辑:徐晓琦

装帧设计:刘 卉

责任校对:周 娟

责任监印:熊庆玉

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)87557437

录 排:武汉正风图文照排中心

印 刷:湖北新华印务有限公司

开 本:880mm×1230mm 1/16

印 张:14

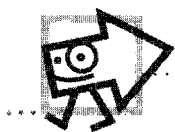
字 数:393 千字

版 次:2010 年 11 月第 1 版第 1 次印刷

定 价:29.80 元(含 1CD)



本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究



序

by Tim Bell

计算机深入到我们生活的各方各面。人们在工作中使用计算机,在家中使用计算机,甚至在上下班的路上也在使用计算机(不要忘记手机和 MP3 播放器也是特殊形式的计算机哦)!但是计算机究竟是如何工作的?它们怎样进行“思考”?人们又如何让计算机运行得更快更好呢?

不同于传统意义上教授“如何使用计算机”的教程,本书着眼于各项计算机技术的原理,它将为你展示真实发生在计算机世界中的点滴,让你理解新技术是如何被设计出来的,并且开发你的“计算思维”来提高解决问题的能力。最重要的是,这本书的目的是让你在学习的过程中也能玩得很开心!

本书作为“不插电的计算机科学”项目^[1]的一部分,旨在让学生们在学习程序设计之前,先来探寻、领悟计算机运作及其解决问题的精妙思想——事实上,在学习上述知识的时候你甚至都不用打开计算机。书中的每项活动都配有详细的解释说明,指出它和相关技术之间的联系以及它如何进一步影响我们的日常生活。这种计算机教学法目前已在世界范围内广泛开展,并被证明相当有效且深受好评。

不用计算机便能学会计算机,是不是让你非常惊讶呢?你能在官方网站 www.csunplugged.org 中找到更多有关“不插电的计算机科学”项目的介绍。

如何使用这本书

你既可以拿此书自学,也可以将它作为信息技术课或数学课的补充读物,在班上和伙伴们一起学习。老师们在指导学生玩游戏时,并不需要深入讲解专业的计算机知识,只要学生稍稍掌握基本的数学技能和拥有一颗学习新知识的心即可完成书中所有活动。我们也特别编写了教师手册部分,里面提供了每道问题的答案和课堂教学的小窍门。

书中的一些小游戏,如果和伙伴们一起来玩会更加有趣!与此同时,我们也提供了自己一个人进行游戏的方案。本书同样也适用于课外兴趣小组或者父母和孩子们进行的亲子活动。

这里采用的教学法能激发学生们依靠自己的力量发现新事物。这是本书一个非常重要的教学技巧,因为本书的主要目的是帮助学生们开发出更好解决问题的方法和自我获取新知识的能力。它将重点强调“计算思维”的培养,而不是单调地学习已有的计算机技术。

随书的光盘中有我们精心准备的视频录像,将帮助你更好地来理解如何玩书中的小游戏。尽管不观看录像也绝对不会影响到我们的学习,不过我们大力推荐在进行游戏之前观看这些录像。录像并不会完整解释书本中的观点,但它们会给你一个直观的印象怎样去做。请记住,只有你亲身参与游戏中时,你才能真正学到东西。

本书含有大量的问答环节,它们是学习过程中的关键部分。这些问题并不是用来测试你对文字本身的理解;相反,它们是你学习的一部分。所以请不要跳过这些问题,而且请确保你能靠自己的力量回答每道问题。如果你被某些问题卡住,你将会在教师手册中找到每道问题的详细解答。

[1] 译者注:“不插电的计算机科学”项目(Computer Science Unplugged)是一个世界范围的信息科学普及项目,它透过一些既有趣又容易的活动,来达到学习“计算机科学”的目的。

在一些小游戏中,你需要准备一些供书写的纸张;如果你不想在书上涂涂画画,可以从随书的光盘或从网站上打印出相关内容。请访问本书的网页 <http://www.hustp.com/forward/toBookPublic.do> 以获取更多丰富的内容和信息。

著名的计算机科学家迪杰斯特拉曾经说过“计算机科学不只是关于计算机,正如天文学不只是关于望远镜”。我们深深希望,通过不依靠计算机来学习计算机能帮助你更好地理解计算机这门科学,拓展你的创造力,帮助你运用计算思维。更重要的是,我们衷心希望你能在游戏的过程中得到乐趣,并能成功完成本书中每一个挑战!

感谢

这个项目得益于许多人的辛劳付出和宝贵建议。

非常感谢华中科技大学出版社对这本书注入的心血和支持,特别要感谢王连弟总编和徐晓琦编辑。来自华中科技大学的刘宏教授和谢夏老师给予我们大量珍贵的建议和支持,对本书的编撰起了关键的作用。和宋恩民教授一起合作,让我们得以反复讨论和试讲本教程。同样,在和来自桂林电子科技大学的董荣胜教授和刘亮龙的讨论中,我们也大有收获。

课堂环节的内容得益于 Robyn Adams 和 Jane McKenzie 发挥的关键作用,他们带来的许多点子都极大地充实了本书内容。

本书中的文字多亏了 Andrew Bell、Anita Dusseau、Stephen Fitchett、Erin Gonzalez、John O'Malley、Pa Kou Vang、Janina Voigt、Phoua Xiong 和 Yuan Wang 的锐眼来校阅。同样要感谢昆明冶金高等专科学校的韩迎春老师对全书细致地审校。

近年来,在许多孩子和教师们的帮助下,我们不断优化项目思路。在中国,我们曾在华师一附中(武汉)、华中科技大学附属中学(武汉)和雷店高中(湖北省英山县)进行了成功的试讲。基督城中学、雪梨小学、尼泊尔小学、西域小学和插云小学(均位于新西兰基督城),舒尔物德山小学(美国威斯康星州麦迪逊市)和南方公园学校(加拿大卑诗省维多利亚市)的小朋友和老师是我们许多小游戏的试验先驱。在此,我们要特别感谢 Liz Edgington、Linda Picciotto、Karen Able、Bryon Porteous、Paul Cathro、Tracy Harrold、Simone Tanoa、Lorraine Woodfield 和 Lynn Atkinson 向我们敞开热情的课堂之门,并对我们的小游戏环节提出珍贵的建设性意见。其中,为我们提供小游戏 9.1 中的照片和天平创意的是嘉田胜先生。

课堂试讲离不开 Gwenda Bensemann、Richard Lynders 和 Sumant Murugesh 的大力支持。有一些小游戏得到了维多利亚市“Mathmania”团队和 Kathy Beveridge 的支持。早期版本的插画由 Malcolm Robinson 和 Gail Williams 绘制提供,其中 Hans Knutson 也贡献了不少宝贵提议。在“不插电的计算机科学”项目的发展中,Matt Powell 为我们提供了无价的支持。特别感谢帮我们测试许多小游戏并提供许多建议的 Paul 和 Ruth Ellen Howard。同样地,Peter Henderson、Ray Hunt、Bruce McKenzie、Joan Mitchell、Nancy Walker-Mitchell、Gwen Stark、Tony Smith、Tim A. H. Bell^[2]、Mike Hallett 和 Harold Thimbleby 提供的大量建议也让我们深受裨益。

感谢我们的家人 Fran、Judith、Pam 和 Remzi 无私的支持,Andrew、Anna A、Anna W、Hannah、Madeline、Max、Michael 和 Nikki 为这个项目^[3]带来许多启发,他们往往是测试各项小游戏的先锋。

我们同样也深深感谢 Google 公司和 Brian Mason 科学技术研究院对该项目的信任和赞助,他们对许多小游戏的开展提供了支持。

非常欢迎你对本书中的游戏提供想法和建议。你能通过 www.csunplugged.org 站点联系到本书的作者们。

[2] 和本书第一作者非同一人。

[3] 事实上,文字压缩的小游戏是由 Michael 发明的。



你将从本书中读到

许多人都知道如何使用信息技术,但是只有少数人理解技术的原理。掌握这些知识,不仅能让你更高效地使用计算机,甚至能让你研发出促进世界发展的新技术。

你或许已经目睹过近几年产生的各种流行电子技术,像搜索引擎、视频网站、网络社区和 MP3 播放器。但是,是谁设计出这些系统并解决了诸多技术难题呢?是计算机科学家、计算机工程师和软件工程师们设计研发出了大多数新技术,你或许能在高等教育阶段学到这些内容,但是我们认为你应该更早接触这些话题。这也是为什么我们出版了这本书。本书中许多章节的内容来自于大学课程,但是我们把它们特别改编成游戏和谜题的形式,从而在你并不是计算机程序高手的时候也能游刃有余地享受学习过程。事实上,你完全可以将计算机放在一边!

这本书还有一个好处:它将开发你的创新思维,使你自己想出有关信息技术的新点子。为计算机界作出巨大贡献的人们都是那些善于解决问题、富有集体合作精神、能不断想出创意的人们。当你还在学生的时候就应该开始这方面的积累,今后或许你就是那个为大家创造新科技从而改变人们生活质量的那个人!

这本书并没有涉及计算机是如何工作的全部细节,但是书中内容能培养你的洞察力。我们主要的目的是揭示那些被计算机科学家和软件工程师运用于设计计算机系统的多元化的思路和技术。当你发现那些看起来难以解决的问题,原来能如此“简单”地被研究人员找到解决途径时,你一定会感到惊讶并产生浓厚兴趣的。

为了满足人们的需要,计算机会在内部处理两类事情。首先,计算机会储存**数据**。数据就是计算机工作时处理的原材料对象(如数字)。计算机会将内部数据转换为人们可以理解的信息(如词语、数字和图像),这是我们将第 1 章到第 7 章重点讲述的内容。

然后,计算机通过执行一系列指令来对数据进行处理。这些指令使得计算机能解决许多问题,例如排序、查找和发送信息。**算法**是一系列解决问题的清晰指令。算法描述了我们是如何让计算机解决问题的,这些我们将在第 8 章到第 15 章为大家具体讲解。

所以接下来……拔掉你的计算机插头吧,让我们一起来享受学习计算机的乐趣!

... Contents
目录

第 1 章 二进制数和“比特”	/(1)
介绍	/(2)
小游戏 1.1 二进制卡片	/(3)
小游戏 1.2 短短的二进制数和长长的二进制数	/(7)
进阶篇	/(8)
有趣的事	/(9)
第 2 章 从小比特到大数字	/(11)
介绍	/(12)
小游戏 2.1 二进制数的性质	/(15)
小游戏 2.2 大一点的二进制数	/(16)
进阶篇	/(20)
有趣的事	/(24)
第 3 章 从比特到字母	/(25)
介绍	/(26)
小游戏 3.1 储藏室谜题	/(29)
小游戏 3.2 制作属于你自己的信息	/(30)
小游戏 3.3 传音游戏	/(31)
进阶篇	/(33)
有趣的事	/(34)
第 4 章 从比特到图像	/(35)
介绍	/(36)
小游戏 4.1 图像解码	/(41)
小游戏 4.2 图像编码	/(43)
进阶篇	/(44)
有趣的事	/(44)
第 5 章 压缩信息	/(47)
介绍	/(48)
小游戏 5.1 文字的解压缩	/(49)

小游戏 5.2 文字的压缩 / (50)

进阶篇 / (53)

有趣的事 / (54)

第 6 章 检测错误 / (55)

介绍 / (56)

小游戏 6.1 翻卡魔术 / (57)

小游戏 6.2 发现更多的错误 / (62)

小游戏 6.3 ISBN 检测 / (66)

进阶篇 / (69)

有趣的事 / (70)

第 7 章 信息 / (71)

介绍 / (72)

小游戏 7.1 Yes/No 问题 / (74)

小游戏 7.2 决策树 / (77)

小游戏 7.3 丢失的文字 / (78)

进阶篇 / (80)

有趣的事 / (81)

第 8 章 程序设计 / (83)

介绍 / (84)

小游戏 8.1 依照指令行事 / (85)

进阶篇 / (89)

有趣的事 / (89)

第 9 章 搜索 / (91)

介绍 / (92)

小游戏 9.1 线性搜索战舰 / (94)

小游戏 9.2 二分法搜索战舰 / (96)

小游戏 9.3 哈希法搜索战舰 / (99)

进阶篇 / (104)

有趣的事 / (105)

第 10 章 排序 / (109)

介绍 / (110)

小游戏 10.1 选择排序 / (111)

小游戏 10.2 插入排序 / (115)

小游戏 10.3 冒泡排序	/(117)
进阶篇	/(118)
有趣的事	/(119)
第 11 章 让排序来得更迅猛吧	/(121)
介绍	/(122)
小游戏 11.1 快速排序	/(122)
小游戏 11.2 归并排序	/(124)
进阶篇	/(126)
有趣的事	/(127)
第 12 章 并行排序	/(129)
介绍	/(130)
小游戏 12.1 排序网络	/(131)
进阶篇	/(135)
有趣的事	/(136)
第 13 章 网络	/(139)
介绍	/(140)
小游戏 13.1 泥泞城市	/(140)
小游戏 13.2 大一点的泥泞城市	/(145)
进阶篇	/(146)
有趣的事	/(147)
第 14 章 路由和死锁	/(149)
介绍	/(150)
小游戏 14.1 橘子游戏	/(151)
进阶篇	/(154)
有趣的事	/(155)
第 15 章 处理输入	/(157)
介绍	/(158)
小游戏 15.1 金银岛	/(158)
小游戏 15.2 用 FSA 来寻找规律	/(163)
进阶篇	/(172)
有趣的事	/(173)
附录 教师手册	/(175)
英文索引	/(209)

第1章

二进制数和“比特”

0

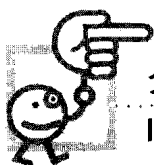
1

Binary Numbers and “Bits”



计算机中的数据无一例外都是以一系列的0和1的形式储存的，但是计算机却能展现形式多样的信息，如文档、网页、照片、音乐、视频等。这是为什么呢？在第1章的小游戏中，我们将来学习如何仅用数字0和1来展现形式各异的信息。

All data in computers is stored and transmitted as a series of zeros and ones. Yet computers can represent a rich range of information: documents, web pages, photographs, music, videos, and so on. How can this be? In these first activities we will begin to learn how different information can be represented using just two different digits: zero and one.



介绍

Introduction

- 计算机功能强大而且十分灵活，人们可以通过计算机浏览网络、发送邮件、观看视频、编辑图片、欣赏音乐、创作小说或者控制机器。但是计算机也有做不到的事情。有时候，人们会抱怨自己的计算机运行得太慢——尽管事实情况是一台价格低廉的计算机也能进行高达每秒数百万次的运算；有时候，想要下载的文件会因为体积过大而难以下载；有时候，计算机还会搞出一堆费钱费时的麻烦事。然而正是因为我们的生活和世界如此依赖计算机，才更应该一同理解它们的优点和不足。



- 对信息做任何有意义的操作前，你需要对其进行保存，以便于第二天想继续工作时还能再找到它。我们将从“数”在计算机中的储存说起，最终我们将学到如何储存文字及图片，如何向小容量的内存空间中储入体积较大的数据，如何防止存储错误的发生，以及如何计算储存了多少信息。

- 首先，让我们来认识一下二进制数(binary number)。所有电子计算机中的数据都是以二进制的形式来储存并传输的，它们直接影响到计算机工作的各项指标：硬盘有多满？下载速度有多快？网上支付有多安全？屏幕可显示的最高分辨率是多少？计算机的最快运算速度是多少？理解了二进制数的工作机制也就意味着你理解了计算机是怎样处理

- Computers are incredibly powerful and flexible. They can be used to browse the internet, send emails, watch videos, edit photos, listen to music, write books and control machines. But they also have limitations. Sometimes people get frustrated with how slow they are—even though a low-cost computer can do billions of operations per second. Sometimes files are too big to download. Sometimes computers cause costly and time-consuming problems. Because so much of our lives and our world depends on these machines, it's important to understand both their power and their limitations.

- Before you can do anything useful with information you have to be able to store it, so that you can read it again tomorrow. We will begin by learning how numbers are stored on computers. Eventually, we will learn about how to store words and pictures, how to store lots of data in a small amount of space, how to prevent errors from happening, and how to measure the amount of information we are storing.

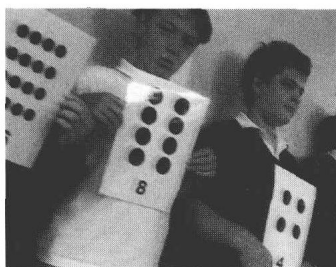
- But first we are going to learn about binary numbers. Because everything on a digital computer is stored and transmitted using binary numbers, they affect many of the things we worry about on computers: how full the disk is, how fast we can download, how secure an online payment is, how high the quality of graphics is on a screen, and how fast the computer can

信息的,许多计算机技术的神秘面纱也会随之被揭开。

很神奇吧,计算机中全部的信息都只用两个数值来储存:0 和 1。这些信息在计算机的内存中以晶体管的开、关状态来表示(开代表 1、关代表 0)。信息还能被储存在磁盘(比如硬盘)、闪卡(比如数码相机的储存闪卡)或者磁带(比如录影带)上,我们将在第 2 章的一组游戏中进一步讲解这些内容,现在主要的任务是来理解计算机如何只用两个数字来表示信息的,即为什么我们称计算机系统为二进制系统(binary 的前缀“bi”代表“两个”的意思,比如自行车 bicycle,双翼机 biplane 或者双筒望远镜 bifocal)。

在我们开始二进制的探秘之旅前,不妨先思考一下,你怎样仅用 0 和 1 来数数呢? 请继续阅读……

视频:请观看二进制数字游戏的录像。



小游戏 1.1 二进制卡片

Activity 1.1 Binary cards

在进行这项小游戏前,你需要准备 5 张卡片,每张卡片画上如下的圆点图案。你也可以用数字替换下面的圆点。

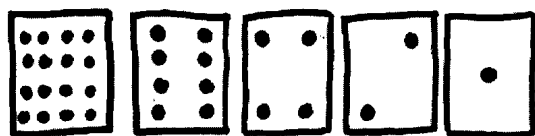
do calculations. Understanding how binary numbers work means that you understand what is happening with the information on your computer in great detail, and it will remove a lot of the mystery of computers.

Amazingly, computers store all information and numbers using only two values: zero and one. This information can be stored in a computer's memory with a transistor that is switched off or on (for zero and one). It can also be stored on magnetic disks (such as a hard disk), flash cards (such as the one in a digital camera), and tapes (such as a digital video tape). We will learn more about this in the next set of activities, but the main thing to understand now is that computers only represent things using two different values, which is why the system is called binary (the prefix “bi-” means two, as in bicycle, biplane, or bifocals).

But first, how can you count using only zero and one? Read on . . .

Video: Watch the Binary Numbers video.

To do this activity, you need to make 5 cards or pieces of paper with dots on them as in the picture below. You can put numbers on the cards instead of dots if you prefer.



1 按照如上顺序,在桌子上摊开卡片。

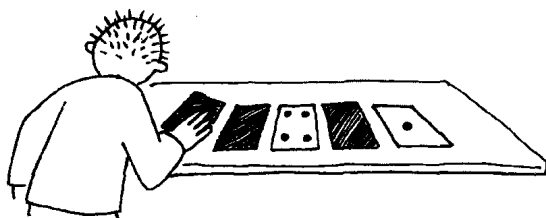
a 你留意到卡片上的圆点图案有怎样的规律了吗?

b 如果我们在这些卡片的左侧增加一张卡片,那么这张卡片上需要画几个圆点呢?

c 如果向左侧继续再增加一张卡片,那么这张新的卡片上需要画几个圆点呢?

d 你是依照怎样的规律来决定刚才新增卡片上的图案的?

2 我们可以将一些卡片翻转使其背面朝上,然后计算正面朝上卡片上点数之和来代表不同的数字。比如,下图中仅将 1 个圆点和 4 个圆点的卡片正面朝上代表了数字 5。这个游戏的规则只有一个:保证卡片要么正面朝上,要么翻过来背面朝上。



e 你需要令哪几张卡片正面朝上来表示数字 6 呢?

f 需要利用哪几张卡片表示数字 20?

g 需要利用哪几张卡片表示数字 15?

h 需要利用哪几张卡片表示数字 21?

1 Put the cards on the table in front of you, in the order shown.

a What do you notice about the number of dots on the cards?

b How many dots would the next card have if we added a 6th card on to the left?

c How many dots would there be if we added yet another card on the left?

d What rule are you using to create the next card?

2 We can use these cards to make numbers by turning some of them face down and adding up the number of dots that are showing. For example, we can make the number 5 by turning over the 1-dot and the 4-dot cards. The rule is that each card is either fully visible, or fully hidden.

e Which cards should you turn face up so that exactly 6 dots are showing?

f How can you make the number 20?

g How can you make the number 15?

h How can you make 21?

 需要利用哪几张卡片表示数字 30?

 任何数字都能用不止一种方法来表示吗? (比如,你能用两种不同的方法来表示数字 5 吗?)

 5 张卡片能表示的最大数字是多少?

 它们能表示的最小数字又是多少呢?

 让我们来试试从 0 数到 31。把全部卡片翻转过来背面朝上代表 0,然后一张一张翻开符合要求的卡片,从 1、2、3、4 一直数到 31。

 在 0 至 31 之中,有你无法表示出来的数字吗?

 令数字递增 1 最简单的方法是什么? (提示:每当数字增加 1,画着 1 个圆点的卡片发生了变化吗? 如果画着 1 个圆点的卡片需要被调整成正面朝下,那么画着 2 个圆点的卡片需要调整朝向吗? 借由这点,相信你能找出将数字递增 1 的简单方法。)

 刚才的卡片游戏利用了二进制数的原理。我们日常生活中用到的都是由 0 到 9 组成的十进制数(decimal number),所有的十进制数都是用 10 个不同的数字组成(1 到 9,再加上一个 0),一旦某一位大于 9 就需要再增加新的位数。比如十进制数 34 就由两个十进制位组成,“3”在这里代表 3 个十。而计算机仅用两个数字 0 和 1 来表示信息,例如刚才使用的卡片,卡片背面朝上代表数字 0,正面朝上代表数字 1。因为只用到两个数值,我们将它们称之为二进制位。5 张卡片可以表示一个 5 位的二进制数。

 由于每位只有两种数值可供选择,所以二进制有时候也被称为基数为 2 的数制(base-two)。

 How can you make 30?

 Is there more than one way to form any number? (For example, are there two ways to make the number 5?)

 What is the biggest number you can make with these five cards?

 What is the smallest number?

 Now we are going to try counting from 0 up to 31. Flip all the cards face down to represent 0, then flip cards over to get 1, then 2, 3, 4, and so on up to 31.

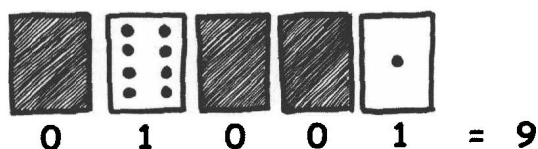
 Is there any number you can't make between the 0 and 31?

 Is there an easy way to count up by 1? (Hint: what happens to the 1-card each time you add one? If the 1-card has to change to facedown, what happens to the 2-card? Based on that, can you find a simple description of how to add 1?)

 The cards that you have used are using binary numbers. Normally we work with **decimal numbers**, which are made up of digits from 0 to 9. There are only ten different digits (1 to 9, plus the 0 digit), but to get numbers higher than 9, we use more digits. For example, the number 34 has two decimal digits, and the 3 actually means 3 lots of 10. On computers the only digits are 0 and 1—there are just two different digits. With the cards, a 0 is when the card is face down, and a 1 is when you can see the dots. Because there are just two values, we call them binary digits. The 5 cards represented a 5-digit binary number.

 Sometimes binary is called “base-two” because there are only two possible values for each digit.

- 比如,二进制数 01001 一共包含 5 位,如果我们用翻转的卡片来表示各位,再对应卡片上的点数,就会得到十进制数 9。下面每张卡片代表一个二进制位。



- For example, the binary number 01001 has five binary digits, and if we turn over the cards to match the digits, we get the decimal number 9. Each of the cards we have used represents a single binary digit.



术语一点通

The phrase “binary digit” is so common in computer science that it is given an abbreviation: “bit”. So a bit is just a digit that can be a zero or one.

在计算机科学中被广泛使用的“二进制位”有一个昵称“比特”(bit)。一个比特即是一个数位,其值可以为 0 或 1。

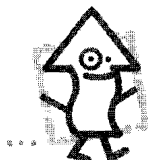
- 这些卡片一共组成多少个比特呢?
- 用卡片摆出二进制数 01101,它对应的十进制数为多少?
- 十进制数是人们常用并熟悉的数字形式,十进制位上的数值范围从 0 到 9,每位上都有 10 个不同数值,因此十进制有时候也被称为基数为 10 的进制(base-ten)。
- How many bits do we have with these cards?
- Use the cards to make the binary number 01101. Which decimal number does it represent?
- Decimal numbers are the types of numbers that people usually use; decimal digits can have the values 0 to 9. Since each digit can have 10 different values, decimal is sometimes called base-ten.

Binary
Decimal

0, 1
0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Base-Two
Base-Ten

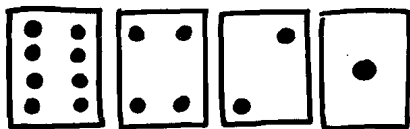
- 二进制数 00110 对应的十进制数是多少?
- 二进制数 01110 对应的十进制数是多少?
- 二进制数 10001 对应的十进制数是多少?
- What is the binary number 00110 in decimal?
- What is the binary number 01110 in decimal?
- What is the binary number 10001 in decimal?



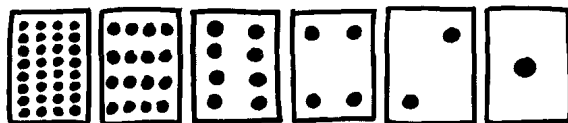
小游戏 1.2 短短的二进制数和长长的二进制数

Activity 1.2 Shorter and longer binary numbers

- ❏ 刚才我们一直用 5 个比特来表示二进制数，但是在现实中往往需要用到各种不同位数的二进制数。例如，我们可以用 5 张卡片将十进制数 9 表示为 01001，而用 4 张卡片将其表示为 1001 也是行得通的。



- ❏ 写下如何用 6 张卡片来表示数字 9。



- ❏ 用以上 6 张卡片最大能表示多少呢？

- ❏ 我们可以用任意的符号来代表二进制中的 0 和 1。接下来不妨挑战一下用不同的符号来表示数字。

- ❏ 算出下面各组符号表示的数值。

$$\begin{array}{c} \boxtimes \boxtimes \boxtimes \boxtimes \boxtimes \\ (\boxtimes=1, \boxtimes=0) \end{array} =$$

$$\begin{array}{c} \uparrow \downarrow \uparrow \\ (\uparrow=1, \downarrow=0) \end{array} =$$

$$\begin{array}{c} \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc \\ (\odot=1, \bigcirc=0) \end{array} =$$

$$\begin{array}{c} \text{☐} \text{☐} \\ (\text{☐}=1, \text{☐}=0) \end{array} =$$

- ❏ So far we've always used 5 bits to represent numbers, but sometimes we will need more, and other times we use fewer. For example, we wrote the number 9 as 01001 using 5 cards, but with 4 cards we only need to write down 1001.

- ❏ Write down the number 9 using 6 cards.

- ❏ What is the largest number that can be represented using 6 cards?

- ❏ We can use any symbols to represent our 0 and 1 in the binary numbers. The following challenge uses all sorts of symbols to represent numbers.

- ❏ Work out the values represented by the symbols in the following chart.

$$\begin{array}{c} + + \times + \\ (+=1, \times=0) \end{array} =$$

$$\begin{array}{c} \cup \cup \cup \cup \cup \\ (\cup=1, \cup=0) \end{array} =$$

$$\begin{array}{c} \blacktriangle \blacktriangledown \blacktriangle \blacktriangledown \blacktriangledown \\ (\blacktriangle=1, \blacktriangledown=0) \end{array} =$$

$$\begin{array}{c} \spadesuit \spadesuit \spadesuit \spadesuit \spadesuit \\ (\spadesuit=1, \spadesuit=0) \end{array} =$$

$$\begin{matrix} \text{☹} = \\ (\text{☺}=1, \text{☹}=0) \end{matrix}$$

$$\begin{matrix} \text{♂♂♂♂} = \\ (\text{♂}=1, \text{♀}=0) \end{matrix}$$

$$\begin{matrix} \text{♂☆☆☆♂☆☆} = \\ (\text{♂}=1, \text{☆}=0) \end{matrix}$$

$$\begin{matrix} \text{☆☆♂☆☆☆☆} = \\ (\text{♂}=1, \text{☆}=0) \end{matrix}$$



进阶篇

Details for experts

❏ (跳过每章的这个小节不会影响你的阅读, 不过这里将谈到一些你或许会感兴趣的内容哦!)

❏ 二进制数中, 任何数位都是其右侧数位的 2 倍, 这就好比在十进制数中, 任何数位都是其右侧数位的 10 倍。比如, 十进制数 423 中, 最左侧的 4 代表 4 的 100 倍, 即 400, 而 2 仅代表 2 的 10 倍, 即 20。比起你熟悉的十进制, 其实二进制一点都不难对不对? 它们仅仅使用了不同的进位基数而已。

❏ 如果二进制数是以 0 开始的, 这些 0 就可以被忽略。例如十进制数 7 可以用 5 位的二进制数 00111 来表示, 它也可以写成 0111 或者 111, 这就像十进制的三位数最大可以到 999, 但是写成 007 的十进制数代表的依旧是数字 7。同理, 00111 只意味着数据以 5 个数位的形式储存, 而拥有 5 个数位的二进制数最大为 11111, 即十进制的 31。

❏ (You don't have to read this section for each topic, but it might contain some information that you were wondering about.)

❏ Each digit in a binary (base-two) number is worth twice as much as the one to the right of it, in the same way that each digit in a decimal (base-ten) number is worth ten times as much as the digit to the right (for example, in the decimal number 423, the left-hand digit is worth 100 times as much as it looks (i. e. 400), while the 2 is only worth 10 times as much as it looks (20)). So the binary number system isn't that different to the numbers that you're familiar with – it just uses a different multiplier for the value of the digits!

❏ If a binary number begins with zeros, they can be left out. For example, the decimal number 7 can be written as the 5-bit number 00111. It can also be written as 0111 or just 111, in the same way that the original number 7 can be written as 007. It's still just the number 7, although having 3 digits implies that the 7 is part of a series that might go up to 999. In a similar way, 00111 means that the number is being stored in 5 bits, and the largest possible number is 11111 (31 in decimal).