



计算机：一部历史

Digitized: The Science of Computers
and How It Shapes Our World

彼得·本特利
Peter J. Bentley



著



顾纹天



译



东西文库

目 录

[内容简介](#)

[作者简介](#)

[中文版序](#)

[引言](#)

[你会计算吗？](#)

[一次性的计算](#)

[二进制数字中的生活](#)

[响彻世界的声音](#)

[计算机把我弄哭了](#)

[构建仿生学大脑](#)

[计算机改变了我的生活](#)

[译后记](#)

书名：计算机：一部历史
作者：[英] 彼得·本特利
译者：顾纹天

东西文库授权青苹果数据中心作全球电子版发行
版权所有·侵权必究

关于东西文库

东西文库（West & East Library）致力于“第三种文化”（The Third Culture）的思考、传播与交流；注重在互联网、科技、商业、媒体、电子阅读等领域的互动；包括但不限于：纸质、电子出版，版权引进、策划，文化论坛。

已出版《失控》、《技术元素》、《字体故事》、《数字乌托邦》、《比特素养》、《变革的力量》等图书。

官方网站：<http://welib.us/>

译文使用权归东西文库独家所有。
未经许可，严禁以任何方式使用。

关注我们：

新浪微博<http://weibo.com/welibrary>

豆瓣小站<http://site.douban.com/151743/>

如有选题推荐、合作意向，请联系：
WElibrary@dongxi.net

内容简介：

《计算机：一部历史》，给大众读者写的计算机科普读物，零门槛入门计算机科学。讲述计算机背后鲜为人知的故事，普及关于计算机和互联网，你不得了解的知识。

在过去数十年里，除非你一直与世隔绝，否则就不可能不受到信息革命的影响。我们身处技术演进史上的计算机时代，无论你是计算机和互联网的拥护者、反对者还是旁观者，无论你是否具备计算机专业背景，只要你使用计算机，这本书就是你的案头必备。

一方面，本书介绍了计算机、互联网领域的奇才，从图灵说起，到克劳德·香农、冯·诺伊曼、恩格尔巴特，再到蒂姆·伯纳斯-李。如果没有他们的启迪，史蒂夫·乔布斯就不可能研制出震惊世界的苹果麦金塔电脑，马克·扎克伯格也不可能创造出一个价值千亿的社交网络帝国。

另一方面，本书勾勒了控制论、人工智能、人机交互界面、密码学、数据库、互联网和万维网问世的整个历史，涉及的前沿领域包括计算机生物、人工神经网络、虚拟现实、仿生计算机、量子计算机等，它们或将引领计算机行业的未来。

对于任何一个想要了解计算机、互联网和科

技发展史的人来说，《计算机：一部历史》都不失为一部简明又生动的读物。

作者简介：

彼得·本特利（Peter J. Bentley），《连线》杂志特约编辑。英国伦敦大学学院（UCL）计算机系教授、荣誉资深研究员。本特利是一个特立独行的计算机科学家，以创造性思维和具有广泛影响力的研究著称。研究领域包括人工智能、演化运算、生物群体智能，以及其他数字科技领域的相关研究。著述颇丰，并被翻译成其他语言广泛传播。著有The Undercover Scientist, Digital Biology等。

中文版序

彼得·本特利 / 文

我祖母活了100岁。她在一生中见证了不少奇迹，比如机动车和客机的问世、家用电器的普及、电视机的发明。我还记得，小的时候，我曾经在祖母那本破旧不堪的词典上查找过“computer”这个词，当时查到的定义是“执行计算工作的人”。

如今，我成为了计算机科学家。每天，我所做的几乎每件事情在100年前都是无法想象的。这里所讲的并不是实验室里的研究工作，而是一些随处可见的寻常小事，比如拨打电话、驾驶汽车、收看提前一天录好的卫星电视节目等等。每天，世界各地数十亿民众的饮食起居都离不开数字计算机，而数字计算机的问世才不过几十年。

千百年来，人类的大脑一直是世界上唯一的计算机。每每想到这一点，就不免令人感慨万千。距今大约2600年前，中国人发明了世界上最早的计算用具——算盘，给人类社会带来了彻底的变革。中国人现在仍然知道算盘在数学教学活动中的重要性，而世界上大多数国家却已将这份宝贵的历史遗产忘得一干二净。算盘之后，计算

尺、对数表⁽¹⁾和简单的机械式加法机相继问世，计算科学在漫长的历史进程中稳步发展，到了近代，却骤然开始突飞猛进。近几十年来，人类社会经历了一场广泛而又深刻的变革。

在本书中，我将给大家介绍一大批引领计算机科技革命的先行者。他们是出类拔萃的天才，拥有超乎寻常的数学能力。是他们提出了二进制信息的思想，奠定了计算机器的理论基础，造就了一个又一个惠及大众的技术奇迹。我很荣幸能够见到他们中的一些人（以及与他们共事的研究人员）。这些改变世界面貌的先行者跟我分享了不少激励人心而引人入胜的故事。同时，我也很荣幸能够采访当今时代引领技术潮流的年轻新秀，聆听他们的精彩故事——其中有不少人白手起家，成为了巨富骄子。显然，无论从事哪一个行业，要想赚取巨额财富，通晓计算机是必不可少的条件。

中国已成为计算业大国。在这个令人惊叹的国度里，数以万计的大小企业每天都在生产各种各样的计算设备和软件。它们雇佣了数百万名员工，产品远销世界各地。2013年，中国国防科技大学研制出了世界上最快的超级计算机——天河二号。

接下来会发生什么？在计算科学界，我已经学会了不做任何预测，因为所有的一切都变化得

太快，让人目不暇接。不过，正如本书所述，如果你洞悉了改善计算机技术的秘诀，那么，下一个计算机先驱很可能就是你。

注释

[\(1\)](#) 对数表是指通过计算得出从1开始各个整数的对数（现在一般用常用对数），所编排成的表格。

引言

它们不厌其烦地执行人的指令；它们收集世间万物的知识，供人顷刻之间随心调取；它们是现代社会的中流砥柱，但其存在却往往备受忽视。

它们是计算机，是人类迄今为止最伟大的发明成就，是登峰造极、至高无上的终极工具。计算机处理器和软件代表着人类最精巧繁复的设计作品。计算机科学的问世，推动了人类历史上最非比寻常的社会变革之一。

启动计算机，打开互联网浏览器，在搜索引擎中输入关键词“披萨”，结果列表中就会显示你家附近的披萨餐厅。鼠标轻轻一点，你就可以输入自家地址，查询店家能否外送了。店家可以外送！而且还支持网上下单。你挑选好自己觉得对味的披萨菜品，配上喜欢的馅料。店家甚至支持网上付款，于是你输入信用卡账号、家庭地址和外送时间，选择“即点即送”，点击“付款”确认。

仅仅过了35分钟，就有人敲响了你的家门。披萨饼送到了，热腾腾的冒着香气，令人垂涎。你给外卖员付了小费，端着披萨饼坐到桌边，开始大快朵颐。

订购披萨饼对很多人来说都不是什么稀奇事。如今，越来越多的餐厅开始为数以百万计的消费者提供味美价廉、品类丰富的餐点和迅捷方便的外送服务。尽管这样说或许令人诧异，不过

如果没有计算机，这一切都不可能发生。在上述情况下，你或许会以为只有一台计算机在运作。但是如果透过现象看本质，你就会发现，为了将披萨饼安全送达，需要动用的计算机数量之多，令人震惊。

事实上，你在启动自己的计算机时，也激活了数量众多的计算机，它们协同工作，使显示屏、鼠标、键盘、宽带和主计算机顺利运转。有了电话公司和网络运营商的计算机牵线搭桥，你的计算机连上了互联网——这是全世界范围的计算机网络。搜索“披萨”时，搜索请求在好几台计算机之间传输，最后才传送到搜索引擎的计算机。这些特殊的机器每天都在刷新互联网页面的数据索引，通过查找关键词，几乎眨眼之间就能给你搜索结果——披萨餐厅的链接列表。点击其中一个链接，你的计算机就能连上餐厅的计算机（同样，这需要动用一系列计算机，只不过和搜索时动用的计算机不同）。餐厅的计算机会将菜品资料传回你的计算机，你选择的披萨饼馅料则会传送过去，以计算价格。点击“付款”时，你的计算机通过安全加密的链接与另一台计算机交换信息，这台计算机则通过更为安全的链接将数据传送到银行，将你的户头资金转账到餐厅的账号中。

支付完成后，连锁餐厅的主计算机将订单发

送到你家附近门店的计算机上。短短30秒的时间里，厨房就收到了你的订单，开始制作披萨饼。这是家上档次的餐厅，所以会现点现做，使用的食材包括玉米粉、小麦粉、番茄酱、马苏里拉奶酪⁽¹⁾、烤胡椒香肠、橄榄油等等。

不过这里面有一个矛盾的地方——虽然菜品非常便宜，但使用的却是优质上等的食材。之所以能够精选上好的意大利食材，原因只可能有一个：库存精度高，可以最大限度地减少浪费；而且进货批量大，可以降低进价。为了做到这一点，门店的计算机每天都会跟踪销量和库存水平（同时结合近两年的数据进行分析），自动生成进货方案，交给经理审核或修改。审核过的进货单随即发送到连锁餐厅的主计算机，主计算机则将所有门店的进货单统合起来，向各大食品供应商下单。一石激起千层浪，餐厅一进货，一连串的计算机开始交换信息，世界各地的配送商、食品加工厂、航运商、食品生产商都开始互动起来。一张披萨饼的食材可能来自世界各地：马苏里拉奶酪产自意大利，烤胡椒香肠产自爱尔兰，玉米粉产自美国。每家公司都依靠计算机维护和打点账目、收入、计划和通讯系统，很多公司都在工厂使用计算机控制的机器进行自动化生产。

你在计算机上完成付款操作后，短短十分钟内，新鲜食材制成的披萨饼就大功告成。厨师将

它放入计算机控制的传送带式烤炉里。烤炉高效节能，烤制迅速。生披萨饼放进去后，只需十分钟，便可完美出炉。外送员随即将烤好的披萨饼放进餐盒中打包好（餐盒在几天前已由计算机控制的纸板切割机提前制好），一次性将好几个餐盒装进特制的保温包内，背着保温包向外送车走去。

由于你选择的餐厅外送范围广，外送员驾驶的交通工具是汽车。和大多数现代汽车一样，这台汽车上也充斥着各式各样的计算机，以确保发动机的高效运转，同时运行牵引力控制系统、防抱死制动系统、车内气候控制系统、乃至车载收音机。电台演播室发送的无线电信号须臾之间就传送了过来，不过，同时向外送车发送无线电信号的，还有另一组计算机控制的通讯设备。这里提到的外送员是个新手，他为了寻找路线，使用了GPS导航设备——GPS导航设备是一台计算机，它能接收轨道卫星的计算机发出的信号（而卫星信号则由更多的计算机校准）。

你所不知道的是，尽管车上搭载了这么先进的设备，外送员还是迷路了。此时距离你付款下单，已经过去了半个小时，外送员开始用手机呼叫餐厅门店，核对你的地址。手机内置的微型计算机发出特殊的无线信号，信号由附近几座蜂窝基地站的接收机接收。控制接收机的计算机互相协

调，同时与外送员的手机交换信息，计算出信号从外送车传递到餐厅门店的最佳路径，并根据每个接收机的信号强度和信道容量随时调整传送路径。正是由于这种动态调整的计算机网络机制，外送员与门店工作人员的通话非常顺利，他发现自己之前把你的地址输错了。

五分钟后，外送员来到你家门口，从保温包里拿出披萨饼，敲响了你家的门。

揭秘计算机

你只要稍微审视一下现代世界中的任何活动，就会发现，在这背后起作用的计算机数量之众，是你未曾想过的。计算机太不显山露水，大多数人完全察觉不到它们的存在。计算机要是有感情，一定会觉得自己备受冷落。它们虽然不易察觉，但并不需要我们大费周章地穿上游猎装、戴上双筒望远镜去荒郊野外仔细寻找——只要把平日里习以为常的活动看得更深刻一点，就能找到计算机的身影。开灯时，屋子里之所以会亮堂起来，就是因为发电站的计算机在正确的时间里接通了正确的电流。打电话给朋友时，你激活了一个非比寻常的计算机网络，网络中的计算机互相交换信息，将你的声音信号传递到朋友的耳中。看电视时，你看到的是计算机实时解码的图像信号，信号由计算机传输，并在计算机上以数

字图像的格式记录下来。计算机是我们无形的助手，它们不事张扬，却近乎面面俱到地服务于我们的衣食住行。

每每想到计算机科学给人带来的巨大影响，我就不由得震惊不已。就在几年前，只有像我这样的计算机科学家才会谈论一些涉及到电子邮件、网页、磁盘分支、操作系统的艰涩术语。如今，我坐火车的时候，身边无论是蹒跚学步的小毛孩，还是耄耋之年的老太太，都在使用这些术语——而且还用对了地方。八卦和时尚杂志如今已开始登载计算机相关的专题文章，这些杂志还可以在平板电脑上在线阅读。曾经只有在计算实验室里才会用到的技术，如今已像燎原的野火一般席卷全世界。试想，倘若类似的变革发生在生物学上，恐怕细胞膜、核糖体、细胞骨架这类复杂的术语都会出现在人们的日常用语中。

这20年可谓沧桑巨变。20年前，我在计算机实验室里第一次看到了可视电话和3D电视的样机，如今它们的技术正成为消费产品的主流。20年前，第一代光学字符识别（OCR）程序还在艰难地读取任何可能识别的字符，如今，15%的出版物已通过OCR技术实现数字化（这一比例还在增长），人们可以从5万亿单词的浩瀚文海中即时搜索某个短语或单词。

互联网问世的情形，我还记忆犹新。第一次

看网页时，完全不懂这样做的意义何在。要等十分钟，网页上才会显示出文本和画面来，何必呢？20年后，我家的电视已经联网，可以播放电视公司提供的在线电影了。手机也可以上网，无论在哪里，都可以查找精确的卫星方位图。我几乎可以从任何国家在线购买任何产品，然后等着派件员送货上门。很难预测在未来的20年里，人类将取得怎样超乎寻常的成就。我的猜测是，未来技术的成本会更低，发展会更迅速，产生的变革会更震撼人心，这一切都将超乎人们的想象。

无论你做什么，都会有意无意地使用到某种类型的计算机，不管是为了工作还是休闲。计算机不仅实用，而且造价低廉，凡是你能想到的任何活动中都能找到它们的身影。但凡以计算机为中心的行业，在不久的将来都是预计增长最快的，只有医疗业的增速可以与之匹敌。增长最迅猛的几个行业在几十年前甚至还没有问世，它们是：信息技术咨询、互联网托管和出版、有线和卫星电视节目、计算机系统设计、软件出版。世界上财力最雄厚的富豪中，有很多从事的是计算机行业，这并不是巧合。当初又有谁能想见，计算机的销量能超过石油、钢铁、粮食乃至时尚服饰呢？不管你的理想志趣如何，要想在现代世界取得成功，懂计算机是必不可少的条件（至少得知道如何高效使用计算机）。

在当今时代，如果一个人说自己是用计算机工作的，那么他的说法毫无意义，因为人人都这样。关键是用计算机从事什么样的职业：是艺人还是音乐人？是金融交易师还是销售员？是产品设计师还是工程师？是外科医生还是护士？是超市助理还是快餐店厨师？是大学生还是中学生？计算机已经渗透到人类生活的方方面面，掌握计算机知识的重要性再怎么强调也不为过。对这种电子机器的运用可能会贯穿于一个人职业生涯的始终（也可能会塑造其子女的生活）。

在本书中，我会跟大家分享计算机科学的奥秘，这些奥秘虽然教科书里不会提及，但却是使计算机大行其道的当今世界顺利运转的关键所在。我们将一起踏上探索的旅程，探访计算机科学家的神秘实验室。我会给大家介绍为数众多的研究人员（还有一、两个白手起家的百万富豪），是他们造就了、并将继续造就我们身边的技术奇迹。在他们的帮助下，我们将见识到，计算机如何在不计其数的领域中得到广泛的运用。

我会向大家展示，计算机从大体上讲非常好学；这方面的知识只要运用得当，就可以使我们在工作和休闲活动中出类拔萃。我们将一起探索这门新兴学科的发展历程：计算机科学萌芽于艾伦·图灵（Alan Turing）等先驱提出的理论概念；到了互联网时期，这一行业迎来了井喷式的增

长；在艰难的成长阶段，人工智能的前景从未实现；此后，互联网泡沫迅速滋长，直至破裂；在最近的发展阶段，计算机科学作为一个（半）成熟的领域，已经能给人类社会带来举世瞩目的成就。在顶尖的研究人员和行业先驱的帮助下，我们将回顾计算机科学史上的成败起伏，探讨在未来的20年里，什么样的技术革新将改变我们的世界。

我是个狂热的向导，因为我自己就是计算机科学家。如果不是从小痴迷于这门学科，长大后绝不可能制造机器人、从事计算机编程。计算机或许是我们无声的奴隶，盲目地执行每一项指令，但是我认为，我们可以从计算机上学到很多东西，通过它们了解这个世界乃至人类自身。

计算机科学

订购披萨饼的故事或许已经帮你从日常生活中不易察觉的角落里找到了计算机的身影。然而，了解了计算机运行的物理设备⁽²⁾和软件，只不过是看到了计算机的冰山一角。硅芯片和程序远远不是计算机的全部。这一点毋庸置疑——要不然这种电子机器也不可能会变得如此普及。

计算机科学是用数学和工程学谱写出来的美妙音乐。正如一个人光会几门乐器和几首流行歌曲，根本不足以领会音乐的博大精深，我们要是

光看几个计算机和软件包的实例，也不可能学会计算机。音乐这门学科历史悠久，传统深厚，百家争鸣，激情勃发。在音乐的世界里，各种理论和想法交流碰撞，表演方法和乐器多种多样，名家大师层出不穷。计算机科学也是如此。这门学科虽然远远不像音乐那般历史悠久，但其发展史依然可以称得上丰富多彩。它形成了自身的传统，也不乏各种思潮的论争和昂扬澎湃的激情。形形色色的理论和想法构成了它的核心，在此基础上派生出各种方法，并通过计算工具加以实践。这一领域也涌现了不少大师，不过他们和音乐家不一样，可能永远不会穿着燕尾服在观众面前表演，也很少受到公众的热烈追捧，尽管他们的成果改变了数百万人的生活。这些大师很可能相貌平平，穿着牛仔裤上班，平常喝了太多的软饮料或咖啡。

你要是向公众询问他们对计算机科学的看法，就会收到很有意思的回复。以下是几个例子：

计算机科学就是与硬件和软件有关的科学。

计算机科学就是捣鼓计算机的科学。

计算机科学主要涉及到编程、工程学和数学。

计算机科学的领域和食品工业一样广泛。

这些回答并没有错，而且读者朋友们可能会认同他们的观点，但是我希望你们在读完整本书以后，会给出稍微不同的答案。

计算机科学是一个神秘的领域，所有的计算机技术均发源于此。从业人员的工作场所往往是高校实验室或者科技公司，但也有一些令人意想不到的地方，比如癌症研究实验室、银行、计算机游戏公司、艺术工作室、汽车制造公司、出版公司等等。计算机科学家几乎可以在任何地方从事任何职业——这也是投身于这一行业的乐趣之一。

计算机科学历史太短，多样性太强，有些人甚至认为，应该将其称为工程学或数学的分支。计算机科学面临着自我定位的问题，或许作为一门专业学科，这一点令人意外。生物学家对自己的专业范围心知肚明，物理学家在这一问题上甚至知道得更精确。但是你要是向计算机科学的领军人物询问他们对这一学科的私人定义，结果就会有所不同。他们的答案五花八门，直观而引人入胜地反映出，计算机科学领域的研发工作门类多样。为了具体说明这一点，我给自己的邮箱联系人、各大社交网站及计算机科学家的聊天群发送了一个问题——“你个人对计算机科学是怎么定义的？”。（你要是想与计算机科学家交流，最好的方法就是使用计算机。）世界各地的计算机科学家都给出了热心的解答，以下仅举出部分例子。有些科学家的名字还会在后文的章节中出现。

安吉拉·萨斯教授来自伦敦大学学院，其研究领域是以人为本的技术和信息安全。她的回答是：“计算机科学是设计和制造特定系统的科学，这些系统功能实用、物有所值，且有利于个人和群体的福祉。”

巴里·费金教授来自美国空军学院（United States Air Force Academy），其研究领域是计算机硬件，包括机器人学。他表示：“计算机科学就是发现和创造特定事物的科学，这些事物要么可以计算，要么应该计算，要么本身就是高效计算的秘诀所在。”

德里克·希尔是IXICO公司的创始人和CEO。IXICO涉足医药行业，专业从事医疗成像服务。希尔的回答是：“计算机科学运用创新技术，挑战悬而未决的科学问题，将新的科学发现转化为改变人们生活的技术。”

佩雷·维勒兹是朴次茅斯大学（University of Portsmouth）的教师，其研究领域是音乐和音效技术。他的回答是：“计算机科学就是用技术引领新的社会风尚。”

迈克尔·菲尔曼是伦敦大学学院的博士，其研究领域是激光测距仪的数据处理。他的回答贴切地概括了计算机科学的多样性：“将医疗影像学、人工智能、生物信息学这类多样的学科统统归类为‘计算机科学’，就因为这些学科刚好需要

用到计算机。这样的归类有时显得毫无逻辑。这就好比把英语、法学和社会学都归类为‘钢笔学’，就因为这些学科都需要用到钢笔。”

我们可以看到，专业人士对计算机科学的定义不仅五花八门，而且显然各有分歧。有些人可能会觉得这种情况很成问题，但事实上，这正是计算机科学大获成功的秘诀所在。实验室里并不像你们想象的那样，聚集了一堆脸上长满粉刺的极客，整天除了写一些只有他们自己才看得懂的代码之外，什么事也不做；能够留在实验室里的，都是世界各地百里挑一的顶尖人才，他们处理的是人类面临的实际问题。这些研究人员有的对癌症数据建模，预测新药对人体的效果；有的利用虚拟现实⁽³⁾帮助恐惧症患者走出阴霾；有的发明新的安全系统，便于人们使用网上银行、遥控车锁和信用卡；有的致力于完善新的医疗成像系统，让扫描的精度更上一层楼；有的专攻移动设别，让无线技术更可靠；有的着力于制造最好的接口⁽⁴⁾或外设⁽⁵⁾，使用户操作起计算机来更加得心应手；还有的构建更完善的金融系统，推动经济的平稳运转。

计算机科学多样性强，而且这种多样性正与日俱增。计算机科学家可以是工程师、数学家或科学家；其专业领域可以是理论、实验或应用；辅助其工作的，可以是硬件、软件或人。但是无

论他们怎么称呼自己，其工作内容是什么，没有他们的共同努力，就不会有造福于人类生活的各项技术。

本书是对计算机科学界的礼赞，这一行业日新月异，精彩多样，硕果累累。我们无法逐一细看每一个领域和每一次革新——我的笔头赶不上它们的发明速度，但是我会向大家展示自己觉得振奋人心的精彩瞬间。希望读者朋友们在这段探索的旅程圆满结束之际，能够理解为什么我等计算机科学家会对计算机如此心醉神迷。计算机科学对人类生活的影响积极、广泛而深远，在这一方面，没有任何技术可以与之匹敌。我们在这段未来主义⁽⁶⁾的旅程中需要探访的第一站，或许可以称得上是这一切的起源，是整个计算机科学的根基所在。我们必须探访支撑这座技术大厦的宏伟支柱，那里是自动计算器的发源地。

计算机的美妙之处并不在于其闪烁的灯光、旋转的磁盘、成排的芯片和电线，而在于，这些部件的背后隐藏着一个优雅而简单的想法。这个想法同晶体管、操作系统、网络 and 文字处理器毫无关联，也不可能有任何关联，因为它的诞生时间比这些设备都早。这一想法是24岁的艾伦·图灵在1936年提出的。

注释

(1) 一种源自于意大利南部城市坎帕尼亚和那不勒斯的淡奶酪。

(2) 物理设备是指硬盘和系统所挂接的各种设备。

(3) 利用电脑模拟产生一个三维空间的虚拟世界，提供使用者关于视觉、听觉、触觉等感官的模拟，让使用者如同身历其境一般，可以及时、没有限制地观察三度空间内的事物。

(4) 指连接两装置的电路，可使数据从一种代码转换成另一种代码。

(5) 外部设备简称“外设”，是指连在计算机主机以外的硬件设备。

(6) 发端于20世纪的艺术思潮，它的一些基本原则包括对陈旧思想的憎恶，和对速度、科技和暴力等元素的狂热喜爱。汽车、飞机、工业化的城镇等等在未来主义者的眼中充满魅力，因为这些象征着人类依靠技术的进步征服了自然。

你会计算吗？

计算机是由数学和工程学领域的先驱在政治动荡和战争时期发明的，它不仅仅是电子机器那么简单。在不计其数的精密电路和软件的背后，闪耀着一种数学的纯粹，那本身便是简洁质朴的美。为计算机的诞生打下根基的数学理论反映了现实世界自身的本质。

如今，挑战不可能之事的科学家依然在竞相探索宇宙的极限。数学和技术领域的革命，是以数百万美元的投资风险换来的，但是谁又能责怪他们呢？

1926年，英国由于煤矿劳资纠纷问题而爆发总罢工⁽¹⁾，汽车和火车全线停运。当时正逢学校开学，14岁的艾伦·图灵要去的是一所精英寄宿学校：位于多塞特郡的谢伯恩男子中学。但是他住在南安普敦，离学校大概有60英里远（约合96.6公里）。很多学生这时候都会干脆在家休息，等着为期十天的罢工结束以后再去上学，这样就可以享受更长的假期了。但图灵不是这种人。他毅然骑上自行车，往学校奔去。他骑了整整两天的时间，中途只在一家小旅馆歇了歇脚。就这样，年轻的图灵准时赶到了新学校。

图灵之所以养成了独立的性格，或许是因为他和哥哥约翰并不在父母身边长大。图灵夫妇都住在印度，但他们认为，孩子应该在英国接受教育。于是，他们把孩子留在了英国，和朋友同住。直到1926年，小图灵的父亲才退休回到英国

——他回来的时候，小图灵正骑着自行车风风火火地往学校赶。

这是一个令人惊叹的开端，但是艾伦·图灵在新学校的成绩并不太好——以前也从来没有好过。他的字迹潦草，英语写作奇差。英语老师给出了这样的评语：“我可以原谅他的写作，但我这辈子都没见过写作水平这么差的。他的作文一向用词不当、粗制滥造、字迹潦草，不管写多少篇都是这样，我已经尽了最大的努力容忍他的劣质作文了……”拉丁语老师的评价也没有好到哪里去：“他成绩落后，总是犯一些滑稽可笑的错误。”

之所以会出现这样的问题，原因是图灵并不在乎课内成绩，而是把时间都花在了自己感兴趣的学问上。他独自开展化学实验，每次遇到数学难题都自己想办法解决。凭借独创的解题方法，他几乎包揽了学校颁发的所有数学奖项。任课老师不知道的是，图灵甚至已经开始从祖父给他的书中学习爱因斯坦的相对论思想和量子力学的最新理论。然而，他的才能并没有得到校长的赏识，校长说：“图灵要是想继续留在公学（英国的私立精英学校），就得让自己变得富有教养。如果他只是想当一名科学专家，那么他在公学上学简直是浪费时间。”

从小时候开始，图灵一直将一头深色的短发

梳成标志性的左偏分。他的声音并没有随着年龄的增长而低沉多少，口吃的毛病也总是让周围的人误以为他不庄重。据图灵的母亲回忆，他在学校里没什么朋友，看起来日子也过得并不快乐。有位同学曾经给他画了张素描，画中的图灵正出神地看着曲棍球场中央生长的一簇雏菊——当时球场上正在比赛。不过，图灵不仅是个梦想家，还是个运动健将。他经常练跑步，长大后还成了训练有素的马拉松运动员。有一次，他在宿舍的楼梯井里自制了一个傅科摆⁽²⁾的复制品，来显示地球的自传。这件事终于使他在学校受到了一些重视。

图灵升上男子寄宿学校的预科班后，遇到了人生中的第一个重要的朋友——克里斯托弗·莫科姆（Christopher Morcom）。莫科姆也是一名天资聪颖的学生。两人经常用化学和数学难题互相挑战。在莫科姆的影响下，图灵申请了剑桥大学三一学院的奖学金。他们一起去面试，结果克里斯托弗顺利通过，而图灵不幸失败。没过多久，悲剧发生了。克里斯托弗患上了牛结核病——这种结核病，可以通过病牛未经巴氏消毒的牛奶传播给人。他没能战胜病魔，英年早逝。18岁的图灵受到了沉重的打击。痛失挚友的创伤使他开始深入思考生命与物理学的关系——他在余生的大部分时间里也在研究这些学科。为了纪念挚友，他

下定决心继续努力，实现两人共同的志愿。因此，到了第二年，他又一次参加了考试，申请了剑桥大学国王学院⁽³⁾。这一次，他成功考取了自己理想的学校。

考上大学后，图灵突然发现自己来到了一个全新的世界。在剑桥，他可以自由探索自己的想法，充分施展打破常规的天性。他开始参与社交，练习划船，同时继续坚持长跑。在学术上，他继续深入钻研量子力学、数学和逻辑学。在道德科学俱乐部（Moral Science Club，剑桥的一个哲学讨论组），他读到了一篇关于数学和逻辑学的论文。同时期的俱乐部成员概括了图灵在这个问题上的观点：“他认为，不能只从逻辑的角度看数学；一个数学命题可以有多种解读方法，逻辑解读只是其中的一种。”换句话说，图灵认为，数学可能比逻辑学更加博大精深。

1934年，图灵以优异的成绩毕业，并继续在剑桥深造，学习数学基础的高级课程。他写了一篇奖学金论文，证明了统计学的中心极限定理⁽⁴⁾——后来发现这个定理在很多年前已经被证明过了。这种事情对于图灵来说已经是家常便饭，而且他这么做是有充分理由的。很多年后，他的同事詹姆斯·威尔金森⁽⁵⁾（James Wilkinson）道破了其中的缘由。“图灵有个强烈的嗜好，他喜欢从最基本的公理出发来推导结

论，他通常只审一遍题，就开始自己想办法解决，完全不参考前人的解法。显然，正是因为养成了这样的习惯，他的解法才那么具有独创性，可以说是自成风格。这让我想起了贝多芬说过的一句名言。当时有人问贝多芬听没听过莫扎特的曲子，毕竟莫扎特正备受关注。贝多芬说，“没有，我也不应该去听，以免受到影响，扼杀自己的创造力。””

“图灵把这个信条贯彻到了极致。老实说，我一开始对他的做法还挺恼火的。每次他给我布置一个任务，我完成以后，他都不肯赏脸看一看我的解法，而是会自己先解一遍；只有自己先初步尝试一遍之后，才会看我的解法。我很快就看到了他这样做的好处。首先，他如果不亲自尝试，是不会轻易接受别人的想法的，不过更重要的是，他经常会想出一些具有独创性的方法。这些方法我可能想都没有想过，而且他要是一开始就看我的解法，也不一定想得出来。”

探索不可能解开的谜题

图灵在剑桥大学修读高级课程期间接触到了一个课题，以此为契机，他将向全世界展示自己的天才。这个课题非常适合图灵，因为它宏大而重要，直击数学的核心，而且尚未被人解决。

开课老师是剑桥大学的著名数学家马克斯·纽

曼（Max Newman）（后来他也成了图灵的挚友和同事）。课程的重点在于探索数学的极限——是不是一切事物、以及任何事物在数学上都是可证明、乃至可计算的？这些令人费解的想法新颖独特，悬而未决，而且令人振奋。数学被认为是宇宙的形式语言^[6]——我们通过这种方式描述万事万物，计算将来会发生的事情。如果没有数学，那么科学、工程学和经济学根本不可能存在。数学上的漏洞一旦被发现，将在很大程度上决定未来哪些事物可以计算、哪些不可以计算。这些想法很快就激发了图灵的想象。

关于数学漏洞的问题，我给大家举个例子。34年前，剑桥有位数学家——伯特兰·罗素

（Bertrand Russell）发现了一个数学漏洞。此前罗素的工作已经取得了巨大的成功——他证明了所有数学问题都可以还原为逻辑问题，也就是说，所有数学发现都可以用逻辑表达式重新写出来。（很多年后，图灵在道德科学俱乐部也做了同样的事情。）这项工作是伟大的，因为它有助于我们了解数学赖以建立的所有基本真理。但是后来，罗素发现了一个问题。他发现了一个悖论——也就是看起来既正确又不正确的论断。数学家经常寻找悖论，因为你如果觉得某件事情既正确又不正确，那么你的想法肯定有漏洞。所以，通过这种方法可以将很多想法证伪。相比之下，

罗素悖论的性质要严重许多，因为它似乎预示着，整个数学体系是有漏洞的。

罗素悖论与理发师悖论很相似。请大家设想一下：

有一位理发师，他只为了不给自己刮脸的人刮脸。那么他给不给自己刮脸呢？

如果他不给自己刮脸，他就属于“不给自己刮脸的人”，他就要给自己刮脸，而如果他给自己刮脸呢？他又属于“给自己刮脸的人”，他就不该给自己刮脸。唯一说得通的解释是，他既给自己刮脸，又不给自己刮脸——但这在逻辑上是不可能的。所以说这是一个悖论。

罗素悖论与之相似，只不过是关于集合的。集合是指具有某种特定性质的事物的总体。下面我给大家简单地分析一下罗素的思路：假设有两个集合，一个是由碗组成的集合，另一是由盘子组成的集合，两者加起来，可以组成一个由碗和盘子的集合组成的集合；如果在此基础上再累加一个由杯子组成的集合，那么就会形成一个由碗、盘子和杯子组成的集合（也就是餐具的集合）。也就是说，“集合”是个有用的数学概念，我们可以把一个集合包含在其他集合当中。很多基本算术运算法则（比如加法、减法）的证明都用到了数字集合的概念，所以说它们是整座数学大厦的基石。罗素认为，有些集合可以同时包含自身，比方说所有非空集合的集合。假设一个集

合包含一切事物，那么任何事物都是该集合的元素。由于该集合内包含元素，其本身不是非空集合，因而肯定包含于非空集合的集合当中。由此可以推出，该集合包含自身，或者用集合论的术语来说，该集合是其自身的子集。

到目前为止，整个推导过程都没什么漏洞。没有出现悖论，只不过有些思想略显怪异。然而，罗素想到了一个非常特殊的集合，这个集合在数学上完全可以接受，但在逻辑上根本说不通。罗素悖论给我们出的难题是：

假设有一个集合A，它的所有子集都具有一个共同的性质P——它们不包含自身。问题是：集合A是否包含自身？

首先，若A包含自身，则A是A的子集，那么A具有性质P，由性质P知A不包含A；其次，若A不包含A，也就是说A具有性质P，而A是由所有具有性质P的集合组成的，所以A包含A。就像理发师悖论一样，唯一说得通的解法是，集合A既包含自身，又不包含自身。这在逻辑上是不可能的。

罗素悖论的提出之所以让数学家如临大敌，是因为它预示着数学的理论基础存在漏洞。几个世纪以来，数学思想和证明无不建立在一系列的基本真理之上。连加法和减法的运算法则都是运用集合和逻辑学加以证明的。但是罗素悖论表

明，任何数学证明都不再可信。人们曾经认为，数学是唯一可能存在绝对真理的领域，就像笛卡尔所信奉的那样，但如今，这样的理念已不再成立。

罗素悖论还只是这一切的开端。1931年，在图灵攻读高级课程的四年前，有位数学家一劳永逸地证明了数学体系必定是不完备的。他的名字叫库尔特·哥德尔（Kurt Gödel）。

哥德尔最杰出的贡献在于提出了哥德尔不完备定理。其中第一条定理或许最为出名，它与一条悖论相似。这条悖论称为“说谎者悖论”。请大家思考一下，下面这句话是对的还是错的？

这句话是错的。

如果这句话是对的，那么它所指的内容必定为真，因此这句话是错的。如果这句话是错的，那么它所指的内容必定为假，因此这句话是对的。哥德尔的第一条定理可以通过类似的方式表述出来：

$G = \text{“本命题不可以由理论} T \text{证明。”}$

如果命题 G 事实上可以由理论 T 证明，则理论 T 中存在一个自相矛盾的定理 G ，既然有自相矛盾的地方，那么理论 T 就是不完备的。也就是说， T 要是完备的理论，就不可以证明 G ，但是这样一来， T 就有证明不了的命题，也称不上是完备的理论了。于是， G 所指的内容就是真的： G 既无

法得到证明，但又是真命题。由此可见，有些事物不管能否得到证明，都可以为真。

这个脑筋急转弯游戏产生了巨大的影响。人们发现，任何事情都无法通过数学加以证明。有些真理则根本无法证实。

这样的结果是毁灭性的。要知道，千百年来，一代又一代数学家孜孜不倦地投身研究工作，就为了建立一个全面而完备的数学体系，在这个体系中，从最基础的公理到最高级、最复杂的证明都可以确凿无误地加以证实。但是如今，哥德尔不完备定理表明，数学家的努力永远没有成功的希望，一个全面而完备的数学体系永远也无法创立。无论数学的理论基础有多么牢固，总会有一些真理永远无法证实。

图灵在学校也学到了一个与此相关的前沿思想。这是由德国数学家大卫·希尔伯特（David Hilbert）在1928年提出的挑战。这项挑战称为“判定问题”（Entscheidungs problem）。希尔伯特想知道的是，一个命题的真假能否自动判定。他的问题是，对于给定的数学语言，有没有什么方法或者程序可以让机器判定某件事情的真假，并将结果显示出来。这样一来，你就可以告诉这台神秘的机器，你输入的语言符合逻辑，让它判断下面这句话是正确还是错误：“如果所有的姐妹都是女性，而莎拉是你的姐妹，那么莎拉是男

性。”于是机器就会稍加思考，然后输出结果“错误”。或者你可以告诉机器，你输入的语言是算术语言，让它判断下面这句话是正确还是错误：“任何大于1的整数都可以通过质数相乘求得。”于是机器又会思考一番，输出结果“正确”。

虽然这听起来颇为实用，但真正的挑战在于：这种自动化的方法或机器是否有可能存在？自动判定简单的句子似乎并不是遥不可及的事情，但如果是用复杂的数学语言写成的高难度句子，是否仍有可能加以判定？这种万能的真理说明者是否真有可能存在？

永不停机的图灵机

图灵把接下来几个月的时间都扑在了这个问题上。年仅23岁的他或许资历尚浅，但他有一颗极富创造力的头脑，很快就想出了一些绝妙的点子。他所遇到的第一个问题，就是如何构想这个神秘的进程或机器。那个时候，电子学还没有创立，世界上最复杂的电气系统是前不久才问世的自动电话交换机——它们体型庞大，足以占满一座宽敞的大厅。当时的机器只能做一件事情，那就是它们被设计出来做的事情。但是希尔伯特提出的挑战是制造一台万能机，这台机器必须通晓任何数学语言，能够看懂人们用数学语言表述出来的任何命题。要做到这一点，它必须能够按照

任何顺序进行任何可能的数学运算，从而给操作者留出充分的余地改变问题，改编机器的程序。

当时没有任何机器能做到这一点，于是图灵构想了一台能做到这一点的机器。他想象的是一台理论计算机。

在1935年，如果你去翻阅词典，查找计算机的定义，就会查到这样的解释：“执行计算工作的人”。年轻的艾伦·图灵所构想出来的机器可以将人们以往用纸笔进行的运算过程全部自动化。这台机器运作起来就像一个玩家在玩棋盘游戏——比如“大富翁”。它设有内存，而内存这个概念放到大富翁游戏棋中，就相当于棋子的位置、棋盘上的房子和玩家的资产。机器可以进入不同的运行状态，就像游戏当中会出现不同的场景。它的状态可以发生改变，好比玩家按照特定的规则推动游戏的进展。它需要来自外部的指令，遵守特定的规则，以改变运行状态，好比玩家掷出骰子以后，如果走到标有“入狱”的棋格，就得把棋子送进“监狱”。但是，与棋盘游戏不同的是，图灵机遵守的规则可以改变。事实上，规则可以由操作者输入，并储存到内存中（这好比我们在棋盘上写下新的规则）。不仅如此，随着机器运行状态的改变，它所遵循的规则还可以进一步发生改变。（好比一个棋格上原本写着“免费停车场”，后来被改成“走进这个格子你就输了”）。

在游戏规则会发生改变的情况下玩棋盘游戏，显然是一个高难度的挑战！但是试想一下，如果可以改变规则，那么整个游戏的性质都有可能发生改变，比如大富翁可以变成蛇梯棋，国际象棋可以变成西洋跳棋。

当然，图灵所指的并不是棋盘游戏。他想象的并不是棋盘，而是一台能从纸带上读取信息的机器。根据即时读取的指令，机器可以将纸带左移、右移，或在纸带上读取信息、输出结果。不过，不管我们打什么样的比方来理解图灵机的运行机制，它的能力是始终如一的。图灵机是一台理论计算机。由于它可以完成任何可能的数学运算，现代计算机能做的事情都难不倒它（只不过它的运算速度要迟缓许多）。

虽然这台奇怪的新机器终究只是纸上谈兵的假想机，但是这已经足够了，因为图灵只是想从理论上解决希尔伯特提出的问题而已。或许颇具讽刺意味的是，图灵虽然提出了关于通用计算机的思想，但却并不急着证明他的机器可以解决判定问题。相反，他想证明判定问题不可能得到解决，进而说明有些问题在数学上根本不可判定。

为了做到这一点，图灵首先假想他的小计算机正在根据纸带上的信号执行一个运算，接着他提出了一个问题：有没有什么方法可以判断这台机器究竟是会陷入死循环，不停地计算下去；还

是会停止计算，给出结果呢？看起来死循环的可能性很大，方法很简单，比方说在纸带A点上写上“移动到B点”，在B点上则写上“移动到A点”。

这个问题放到今天也很有现实意义，因为计算机一旦陷入死循环，或许就会“死机”，什么事情也做不了，这是我们不希望看到的。好比我们在自动取款机上输入PIN码（个人识别密码）以后，机器应该吐钱出来，而不是一动不动，什么反应也没有！

图灵认为，要想判断他的机器会不会停机，那就需要再构造一台图灵机，以对第一台机器进行检测，因为他知道，他假想的机器在理论上可以进行任何数学运算。于是他假想出第二台图灵机，如果检测到第一台图灵机永不停机，那么第二台机器就会停机，然后输出“不停机”；如果检测到第一台图灵机停了机，那么第二台机器就会一直运转下去。

现在，脑筋急转弯的地方来了。假如第二台机器反观自身，判断自己会不会停止计算，那会发生什么情况？图灵对此进行了设想，他突然发现了一个悖论：如果机器检测到自己会永不停机，那么它就会停机，然后输出“不停机”；如果机器检测到自己停了机，那么它就会一直运转下去。这在逻辑上是不可能的，由此证明，有些图灵机是不可判定的——我们永远也无法判断它们

会不会停机。

尽管这样说或许令人费解、甚至不可思议，但是不可判定或不可计算的问题的确大量存在——自此之后，这样的事实一直让计算机程序员备受困扰。图灵的研究结果表明，有些数学问题是计算机无法解决的，这与计算机的运算能力、运算速度和内存容量无关。

1936年，正当年轻的图灵准备将这个振奋人心的成果公诸于世时，他偶然读到了美国数学家阿隆佐·邱奇（Alonzo Church）前不久发表的论文。那时候，全世界有好多数学家正在着手解决希尔伯特的判定问题。其中有的数学家——比如哥德尔——已经开始有了重要的研究成果。邱奇采用的方法与图灵截然不同，他需要创立新的数学概念和语言，以表述有关函数和演算过程的思想。他使用了自己创立的新语言——称为 λ 演算，并在哥德尔的基础上扩展了研究范围。研究表明，没有任何通用的算法可以判定任意两个 λ 表达式是否相等。也就是说，有些事情永远无法用数学方法加以判定——要想解决判定问题是不可能的。邱奇就这样率先攻克了希尔伯特挑战，他发表研究成果的时间只比图灵早了几个星期。

接下来几年里，还有其他描述算法的理论被相继提出，但它们都是等价的。邱奇的 λ 演算是

经典的理论，它已成为计算机科学的宝贵工具，可用于对软件问题进行形式证明⁽⁷⁾。时至今日，它的地位依然举足轻重。不过，图灵机显然是概念方面的赢家。或许正是因为简洁易懂，图灵的计算机思想已成为理论计算机科学的基础。时至今日，连“可计算性”的定义都是根据他的思想界定出来的。“邱奇-图灵论题”（Church-Turing thesis）得到了广泛接受，该论题认为，任何可计算的问题都可以由图灵机计算。

图灵的贡献

1936年，由于志同道合，图灵决定去美国普林斯顿大学投奔邱奇，他在那里师从邱奇，完成了博士学业。

阿隆佐·邱奇自己的经历就带有传奇色彩。他在高中时由于气枪事故，导致一只眼睛失明，后来又因为这只眼睛失明而不小心被有轨电车撞倒，因缘际会，与照看他的护士坠入爱河，步入婚姻的殿堂。邱奇平时为人彬彬有礼，衣着干净整洁，宗教信仰坚定不移，有一些出了名的怪癖，喜欢阅读和收藏科幻小说，如果发现书中有错误，他会在目录页用铅笔修改，或者致信作者予以纠正。每次讲座开始之前，他都会按部就班地把黑板擦得纤尘不染，擦拭次数非得是偶数，而且一般都要用到肥皂和水。擦完黑板后，他会

耐心地等待水迹风干，要不然不会开讲。每次开讲都是长篇大论，好像在看书稿直接念一样。如果被人打断，他会很不自然地停下来。平时说话很少不用逻辑论证。有传言说，邱奇连吃早饭的方式都很有逻辑：“先把牛奶倒进空碗里，放适量的糖，用早餐勺搅拌均匀，然后放一两勺麦片。吃完这点麦片后，再接着放一两勺，边吃边放。这样一来，糖就会在牛奶中充分溶解，分布均匀，而且麦片也不会泡得太软。”

图灵从来没有变得像邱奇那么爱干净，不过他也培养了一些高度逻辑化的习惯，而且这些习惯有时显得很古怪。博士毕业回国后，图灵喜欢戴着防毒面罩骑车，以预防花粉症。如果他发现自行车经常在他踩14圈以后掉链子，他就会每踩完13圈以后下车调整链子。

1938年，图灵回国后不久，便受到政府代码及加密学校[\(8\)](#)（Government Code and Cypher School）的邀请，协助军方破解德国的著名密码系统——“恩尼格玛”[\(9\)](#)（Enigma）。1939年英国宣战后，图灵开始在这家位于布莱切利公园的密码破译机构全职工作。他很喜欢这份工作，因为它充满挑战，而且工作环境又好。1940年，他发明了一台破译机——名为“炸弹机”（Bombe，得名于波兰的一台破译机），成功破解了德国空军传递的所有“恩尼格玛”加密情报。到了1941年

年中，德国海军的“恩尼格玛”加密情报也被全部破译。1942年至1943年3月，图灵在美国协助破译工作；尽管德军升级了密码，但事实证明，图灵的思想又一次成为了最得力的破译工具。

据估计，图灵的贡献使密码破译工作缩短了两年的时间。这份功劳可谓功德无量，要知道，战争期间每年就有1,100万人死亡。据说温斯顿·丘吉尔曾盛赞图灵，说他的工作为二战的胜利做出了最杰出的个人贡献。对于一位性格略有些古怪的极客来说，这已经是很高的赞誉了！图灵因为二战时期的杰出功劳获得了英王授予的不列颠帝国勋章（OBE），但是由于情报工作的保密性，他的功劳在接下来的三十年里一直不为人知。

二战结束后，图灵（如图1）继续发挥自己的才思，开展富有开创性的研究工作。他构想了世界上最早的计算机之一——自动计算机

（Automatic Computing Engine，简称ACE）。当他知道恩师马克斯·纽曼当上曼彻斯特大学

（Manchester University）教授后，自己也进入了曼彻斯特大学，担任讲师的工作。任教期间，他一方面继续开展数学研究工作，另一方面扩展了兴趣范围，研究了神经科学、个体发生学^{[\(10\)](#)}和量子理论。图灵是人工智能研究领域的先驱之一（我们会在后面的章节提到他的好几个思想）。

1951年，他因为图灵机的研究工作而当选为伦敦

皇家学会（Royal Society of London）的成员。他的大学同事并不知道，他在任教期间依然供职于英国通信总部（GCHQ，相当于国家安全局），继续参与密码破译的工作，直到冷战爆发。



图1 艾伦·图灵（左一，脚踩汽车台阶者）与沃尔顿竞技俱乐部（Walton Athletic Club）成员，1946年摄

1952年，图灵向警方报告了一起入室盗窃案。他承认，行窃的人正是他以前的同性伴侣的朋友。但是同性恋在当时是非法的，因此图灵被控以严重猥亵罪而遭到起诉。审讯期间，他的老朋友马克斯·纽曼为他出庭作证，但是于事无补。图灵不幸被定罪，他的选择只有两个，要么坐

牢，要么接受“化学阉割”——注射女性荷尔蒙（雌激素）。他选择了注射雌激素，为此不得不遭受乳房发育等药物副作用的伤害。他不再具备参与保密工作的资格，不能再为英国政府通信总局工作。他的一举一动——无论是外出度假，还是与外国科学家开展合作——都在国安人员的密切监视之下。

图灵继续开展研究，发表了更多关于个体发生学、量子理论和相对论的论文。他四处旅行，游历了巴黎、雅典和科孚岛（Corfu，位于希腊西北部，爱奥尼亚群岛之一）。但他并不快乐。1954年，图灵被发现死在家中，身边放着咬了一口的苹果。此前他一直在做电解实验，身上可能残留了化学物质，而苹果表面检测出了氰化钾。图灵的母亲和他在通信总部的几名同事认为，这是一起事故。警方的调查结论称，他的死因是自杀。

图灵的伟大贡献从未被世人遗忘。他被大多数学术界人士尊为计算机科学之父。但是公众对他的看法只能用莫衷一是来形容。1998年，图灵故居的门边挂上了蓝牌（Blue Plaque，向名人故居发放的一种证明标志），时值这位伟人的86周年诞辰。当时，罗杰·彭罗斯爵士⁽¹¹⁾（Roger Penrose）写道：“开创计算机革命的中心人物是艾伦·图灵，他杰出的创见和视野使这一切成为了

可能。变革始于20世纪30年代，尽管我们现在难以预见计算机革命的极限究竟何在，但是图灵本人指出了这种理论局限性的存在。”2001年，曼彻斯特的萨克维尔公园（Sackville Park）树起了一座纪念雕像。雕像中的图灵坐在长凳上，手里拿着一个苹果。

如今，英国国家计算博物馆（National Museum of Computing）就坐落在布莱切利公园。馆内设施完全对公众开放，其中包括图灵的办公小屋。图灵的马克杯至今依然拴在小屋的暖气片上（谁也不知道他为什么喜欢把杯子拴在上面）。最近，国家计算博物馆获得了珍贵的图灵论文手稿，作为馆藏展品。苏·布莱克是一名计算机科学家，她经常代表馆方在电视、电台和社交媒体上发起宣传活动。通过布莱克及其他热心人士的努力，国家计算博物馆得到了足够的宣传和谷歌等企业的赞助，可以保持对公众开放。

一说起图灵，布莱克就激动万分。“他饱受迫害，遭到起诉，被迫在坐牢和化学阉割之间做出选择——结果不得已选择了化学阉割……这都是些何等非人的对待啊！他是上个世纪最伟大的人物之一，但却沦落得那么凄惨。”

“我们无法让时光倒流，改变图灵境遇，”布莱克表示，“但是我们可以纪念他取得的伟大成就。希望他的故事可以激励更多的人勇于

奋进，即便他们一开始默默无闻。”

2009年，由于科学家和公众的广泛支持，一项要求政府道歉的请愿活动取得了成功。首相戈登·布朗（Gordon Brown）发表了一篇情真意切的致歉信，信的结尾写道：“……我谨代表英国政府、以及所有因图灵的工作而自由生活的人，对他说：‘我们很抱歉，你本应该得到更好的对待’。”

复杂即是简单

通用图灵机（Universal Turing Machine）已成为世界上所有电子计算机的理论蓝图。它反映了计算机的行为模式，指导着人们设计和制造真正的计算机。正因为有了这个理论基础，我们清楚地知道，任何一台计算机都可以模拟其他计算机的行为（只要有充足的时间和内存）。这一点甚至在第一台电子计算机问世之前就已经为人所知。

图灵机仍然是理解可计算和不可计算问题的最佳理论工具之一。有些计算机科学家认为，这两种类型的事物或许有助于我们深入了解宇宙的本质。伦敦大学的马克·赫布斯特（Mark Herbster）就是这样一位研究人员。“我认为可计算性理论比很多物理定律更有说服力，”他表示，“你可以研究可计算事物和不可计算事物的

基础。它们是现实世界的基本范畴。我认为它们是我们的宇宙乃至一切可能存在的宇宙所固有的事物。”

可计算理论还有一些非常实际的应用：比如计算完成某个运算需要的时间。即便有些运算过程在理论上是可能的，这也并不意味着我们制造的计算机具备与之匹配的运算能力。

理论计算机科学家罗宾·赫希（Robin Hirsch）研究了这些思想。“这个世界上存在三种类型的事物，”他说，“第一种在理论上可能做到，但却无法解决；第二种在实际上可能做到（因此在理论上也必定可能做到）；第三种在理论上可能做到，但在实际上却未必可能做到——虽然说是在理论上可能做到，但往往是比较匪夷所思的类型，即使宇宙的寿命终结也不一定能够完成，所以从实际角度讲，我们也解决不了。在计算机科学领域，大多数有趣的问题都属于这个范畴。”

这是一个非比寻常的思想。无论我们的技术发展多么先进的地步，这三种类型的事物会一直存在。像图灵的停机问题这种无法判定的问题就属于第一个范畴。要想解决它们是不可能的，不管你使用的是什么样的计算机。第二种类型——也就是在实际上可能解决的问题，一般都很容易证明。比方说，文字处理器、电子制表软

件、计算机游戏——这些东西显然可以制造出来，因此，用计算机运行它们肯定是有可能的。但是第三种类型的问题挑战着人类的极限，解决起来要困难许多。有时候这类问题或许看起来不切实际，但我们只是在尝试突破当前的极限，对它们进行运算。有些事情或许根本就不可能实现——我们或许永远也无法制造出内存足够大、运算速度足够快的计算机来解决这些问题，但是不试试怎么知道呢？我们不想把时间浪费在解决不可能解决的问题上，但我们很乐意花时间提高自己的能力，对难度极大的问题发起挑战。

问题的难度一般随着规模的大小而发生改变。假设你是一个较真的学校教师，非得让班里30名学生按照从矮到高的顺序排成一行横队，最矮的站在左边，最高的站在右边。为了告诉学生应该怎么站队，你需要拿着卷尺两个两个地比对他们的身高，每比对一次可能要花上一、两分钟的时间。你不想把一整天都搭进去，所以比对的次数越少越好。那么你需要比对多少次呢？

当然，这个问题的答案取决于学生一开始是怎么站队的。在理想情况下，如果他们刚好按照从矮到高的顺序排成了一行横队，那么你只需要做29次比对。（别忘了，你是一个较真的老师，需要一个个进行检查。）在一般情况下，如果你采用了聪明的排序方法，那么平均需要比对的次

数大约为44次。在最坏的情况下，如果你不幸采用了比较愚蠢的方法，把每个学生都和其他学生比对了一遍，那么你需要比对的次数为 $30 \times 29 = 870$ 次。（如果每一次比对都花了一分钟的时间，那么整个过程下来，就需要耗费14.5个小时。）

在计算机科学领域，我们常用的词是“算法”，而不是“方法”。这个术语主要用于描述我们打算写入计算机程序的方法或过程，它给出了程序运行的所有步骤。因此，所谓排序算法，就是指为一系列事物排序的方法。插入排序是其中一种算法：先让一名学生出列，然后让另一名学生和他排好队。这样一来，班里的学生就分成了两个横队，其中一个是有顺序列，另一个是无序队列。将无序队列中的学生逐一插入到有序队列当中的合适位置，直到所有的学生都排成了有序队列。这种算法并不是很快，但是行之有效。还有一种算法称为抢椅子（只不过不需要真的把椅子用上）：让所有学生都跑动起来，随机改变站队位置，然后喊停，看队列是否有序。如果有序，则大功告成。如果无序，则继续抢椅子，直到所有学生都站成有序队列。这是一个非常愚蠢的算法，因为在最坏的情况下，这样算下去可能会没完没了，学生可能永远也站不好队。

因此，一个愚蠢的算法可能会让你白白浪费

很多时间。相比之下，一个高效的算法运行起来就会快捷很多。当然，所有排序算法的运行时间都取决于排序对象的数量，但是愚蠢的排序算法耗费的时间通常要比高效的算法漫长许多。显然，我们更愿意采用寻找高效的算法——也就是时间复杂度[\(12\)](#)较低的算法。

计算机科学家有一种通俗直观的方法来表述算法的时间复杂度。我们称之为“大O符号”（Big O notation，一种描述函数渐近行为的数学符号）。假设排序对象的数量为 n （在上面所举的例子中， n 的数值为30），如果完成排序的时间仅仅取决于 n 的数值，那么算法的时间复杂度为 $O(n)$ ；如果你把每一个排序对象都与其他对象进行比较，那么算法的时间复杂度为 $O(n \times (n-1))$ ——在这里，我们一般会把常数项和低阶项忽略不计，因为当 n 的数值非常大时，低阶项和常数对结果的影响就会微乎其微，所以， $O(n \times (n-1))$ 可以简化为 $O(n^2)$ 。

目前最佳的排序算法平均时间复杂度为 $O(n \log n)$ ，比 $O(n^2)$ 要好很多。你可以自己检验一下。最快的排序算法称为“快速排序”（Quicksort），它的平均时间复杂度为 $O(n \log n)$ 。有一个较慢的算法称为“冒泡排序”（Bubblesort），它的平均时间复杂度为 $O(n^2)$ 。假设每一次比对需要一分钟，如果排

序对象为十个，那么快速排序需要十分钟左右，冒泡排序则需要一个半小时以上；如果排序对象为一百个，那么快速排序需要3小时20分钟左右，冒泡排序则需要将近一个星期；如果排序对象增加到一千个，那么快速排序需要两天的时间，冒泡排序则需要将近两年！

冒泡排序虽然听起来很糟糕，但它还不是最差的。如果我们采用时间复杂度为 $O(2n^2)$ 的“慢速排序”（Slowsort）算法，那么得出的结果就真的会变得很吓人。算法的运行时间呈指数式的飙涨。排序对象只不过才十个，慢速排序算法就需要耗费2411,816,210,479,888,000,000,000年的时间——这个运行时间甚至达到了宇宙年龄的1,754,050亿倍。

就算我们把比对速度加快（或许得使用某种超级并行计算机），将比对时间从一分钟缩短到理论上可测量的最小值——一个普朗克时间（Planck time）（ 5.316×10^{-44} 秒），一旦排列对象超过30个，慢速算法的运行时间依然漫长得吓人。另一种提升速度的方法就是扩大内存——但是这样一来，空间复杂度⁽¹³⁾就会直线上升：到时候，运行慢速算法需要的内存量可能会比全宇宙物质的量还要多。

需要指出的是，算法之所以在理论计算机科学领域举足轻重，不仅仅是因为我们可以借此让

软件运行得更高效，而且还因为它能够告诉我们，从实际角度看，哪些问题容易解决，而哪些问题难以攻克。排序是计算机能够解决的一个非常简单的问题，就算排序对象的数量级很大也是如此，因为我们能够设计出时间复杂度较小的算法，比如快速排序算法。但是对于有些问题，我们所能取得的最好成果，也就只有慢速排序那点程度了。

至少我们目前是这么认为的。

P是否等于NP？

2000年，美国商人兰顿·T·克雷（Landon T. Clay）公布了千禧年大奖难题（又称世界七大数学难题）。解开任何一道难题的人都可以获得一百万美元的奖金，这笔奖金将由克雷在美国马塞诸塞州剑桥市新开设的克雷数学研究所（Clay Mathematics Institute，简称CMI）颁发。十年过去了，只有一道难题被人攻克，解答者是俄罗斯圣彼得堡的数学家格里戈里·佩雷尔曼（Grigori Perelman）博士。尽管佩雷尔曼为他的研究领域做出了“无与伦比的巨大贡献”，他依然将一百万美元的奖金拒之门外（还有好几家久负盛名的机构想给他颁奖，也遭到拒绝）。尽管如此，学术界还是有不少人对解题拿奖心驰神往。

在尚未解决的六大难题中，有一道题或许是

最简洁的。它的题干是：P是否等于NP？

P和NP指的是两种类型的问题，它们的计算复杂度各不相同。P类问题可以通过多项式时间 [\(14\)](#) 算法解决。换句话说，凡是可以用 $O(n^x)$ 算法解决的问题都是P类问题，不管这里的x是什么。排序问题就是典型的P类问题。就算是最好的排序算法，它的时间复杂度在最坏的情况下也是 $O(n^2)$ ，符合多项式关系，因此排序问题属于P类问题。

对于NP类问题，我们可以在多项式时间内检验候选解是否正确，但是求解所需要的时间却会漫长许多——而且往往是指数时间 [\(15\)](#)。至于这段时间可以漫长到什么地步，大家看了上一节那个慢速排序的例子，想必已经很清楚了。

求解的时间比检验解法的时间还长，这听起来或许有些匪夷所思，但是大家只要思考一下下面这个问题，就会明白其中的道理了。假设你负责给无家可归的人提供福利住房。公寓楼里可以安排一百个人入住。社会福利局给了你一张长长的名单，上面成对地列出了一些彼此合不来的人，他们不能安排在同一栋楼里居住。申请入住的一共有四百号人，你该如何做出选择？

无论你对这个问题如何求解，都可以轻而易举地检验出解法是否正确。你唯一需要做的，就是选出一百个人，然后对照一下社会福利局给的

名单，确保彼此合不来的人没有同时入选就可以了。但是，要选出这一百个人其实非常困难。从四百名申请者当中选出一百个人，这样的组合可以有很多种。在人类已知的宇宙中，原子的数量都没有这么多。因此，这个问题属于NP类问题。我们可以很快地检验出候选解是否正确，但要找到完美的解法比登天还难。

我们发现，日常生活中有很多例子都属于NP类问题。就拿大家都很熟悉的扫雷游戏来说吧，它的玩法就是一个NP类问题。再举几个例子，怎样计算配送车穿行于各个城市的最佳路径，同时尽量缩短车程？这是一个NP类问题；如果你有很多大小不同的行李，怎样装箱才能最大程度地节省空间？这也是一个NP类问题；如果你有一张清单上列出了所有需要完成的家务活，怎样才能有限的时间里安排它们的先后顺序？这又是一个NP类问题；给出一个固定的金额，怎样才能凑够这笔钱，同时尽量少用硬币？就连这种事情也是NP类问题。在已知量较小的情况下，所有这些问题乍看之下都很好解决，但是，一旦已知量的数量级增大，比如配送车穿行的城市增加到一百个、装箱的行李数量增加到五百个、硬币的数量限制增加到一百个，那么求解所需要的时间就会呈指数式增长。

因此，P是否等于NP的问题实际上是在问：

难度很大的NP类问题究竟能否用多项式算法求解？我们现在采用的算法是不是太愚蠢了，就像慢速排序那样？能不能找到像快速排序这种聪明的解法，让原本很难的问题一下子迎刃而解？

多年来，为数众多的计算机科学家一直在努力证明或者证伪 $P=NP$ 。巧合的是，其中一名科学家恰好是我在伦敦大学学院的同事。我第一次见到他的时候，他还是个学生。他的名字叫丹尼尔·赫尔姆（Daniel Hulme）。

早在学生时代，丹尼尔就对 $P=NP$ 问题产生了兴趣，他利用业余时间想出了一个聪明的算法。他把一般用于满足约束条件的方法用在了求解上，从而在多项式时间内解决了一些NP类问题。在这里，他主要研究的是布尔可满足性问题（Boolean Satisfiability，简称SAT），也就是所谓的“判定问题”，只不过是用逻辑运算符“与”、“或”、“非”构建出来的（相当于把电子电路转化为数学问题，我们在下一章还会提到）。布尔可满足性问题的一大应用就是电路验证。这里的判定问题或可表述为：是否任何一组输入都能让电路工作？或者说，是否存在一组特定的输入，使电路无法工作？

布尔可满足性问题属于NP困难问题（NP-hard）。也就是说，它们的难度不亚于NP类问题中最困难的范畴。同时，布尔可满足性问题也属

于NP完全问题（NP-complete）。也就是说，只要你为这些问题找到了多项式时间解法，那么你就掌握了快速解决所有NP类问题的钥匙，因为所有NP类问题都可以归结为某种形式的布尔可满足性问题。如果丹尼尔的聪明算法能够在多项式时间内解决所有布尔可满足性问题，那么他就相当于证明了 $P=NP$ 。

丹尼尔的故事非常引人入胜。“一开始，我的博士论文研究的课题是，如何通过计算机建模，分析熊蜂的视觉机制，”他说，“不过我喜欢在业余时间研究自己感兴趣的 $P=NP$ 问题。我试图说服自己 P 不等于 NP 。这样做很傻，我知道。但是我喜欢给自己挑战！我得出了一些很有意思的结果，导师允许我把这一领域的研究工作继续开展下去，作为博士论文的课题。我想出了一些自己觉得比较高明的实用算法，它们的运算效果应该会比已有的算法更好。”

“我选了一些已经解决的布尔可满足性问题，用自己的算法解了一遍，发现它比很多已有的解法更好。很多其他算法无法解决的问题我都解决了。于是我找到导师，跟他说，‘老师您看，我想出了一个实用的多项式算法，它可以解决这些问题。您能不能帮我看看这个算法是否可行？’”

丹尼尔的导师伯纳德·巴克斯顿（Bernard

Buxton) 将这个问题交给了罗宾·赫希。赫希回忆道,“他觉得自己证明了 $P=NP$, 于是伯纳德让我帮忙看看。经过一番辩论, 我还是发现了一些很难察觉的问题。丹尼尔并没有证明 $P=NP$, 他的算法并不一定适用于所有情况。但是丹尼尔不太相信我。”

这样的消息对于丹尼尔来说有些难以接受。“那时候, 我发现自己的多项式解法不仅能解开一些比较重要的布尔可满足性问题, 而且比很多已有的解法更快。我感到非常振奋, 因为这个基本的多项式算法可以打败布尔可满足性问题的众多解法。我觉得这就是我们以后应该遵循的研究方向。”

罗宾不得不向丹尼尔证明他的算法并不是那么的天衣无缝。“我不得不构建一个病态[\(16\)](#)反例, 这对我来说很难, 但我还是想办法做到了。丹尼尔的算法虽然有局限性, 但却是很好的研究成果。它已经非常接近正确答案了, 在布尔可满足性问题的所有解法中, 我觉得它可以称得上是数一数二的。丹尼尔的研究工作很了不起。”

当然, 这样的结果对于丹尼尔来说无疑是令人失望的, 不过丹尼尔意识到自己的算法依然非常有用。“罗宾的例子表明, 如果问题增多到一定的程度, 我的算法就不适用了。也就是在这时候, 我意识到自己建立了一个很好的预处理算

法。只不过它不是 $P=NP$ 问题的正解——真是太可惜了！”

丹尼尔由于他在 $N=NP$ 问题上的研究成果而被授予了博士学位。毕业后，他在硅谷创立了一家公司，命名为Satalia（这个名称由两个单词组成，它们分别是SAT和alia。SAT代表布尔可满足性问题，alia是拉丁文单词，意为“不同”或“其他”）。Satalia在英国的分公司命名为“NP完全”（NP-Complete）。公司成立的宗旨，就是为数学界找出NP完全问题的高明解法。但是除了这些商业目标，丹尼尔依然对解开“ P 是否等于 NP ”这个的神秘谜题心驰神往。“我觉得人的大脑就像计算机，”他说，“人的大脑经过了漫长的进化，尽管未必达到了最优的状态，但说不定就能找到 $P=NP$ 问题的正解，谁知道呢？或许问题的答案已经通过硬编码⁽¹⁷⁾写在大脑里了，只不过我们还没发现，总有一天我们会把它破解出来的。”

“我的个人理想就是赚足够的钱给自己买一间办公室，再在办公室里配上一块白板（白色的金属板材料制的书写平面），这样我就可以随心所欲地研究 $P=NP$ 问题了。”

预言及其他复杂性问题

显然，NP类问题很难解决，但早在1936年，

图灵对如此艰涩的难题就已经有了自己的思考方法。思考NP类问题的方法之一，就是构造一台特殊的图灵机，称为非确定型图灵机（non-deterministic Turing Machine）。如果我们可以制造出这样的机器，就可以让它在多项式时间内运行NP类问题。之所以称之为非确定型，就是因为我们无法预测它的运作方式，但它总能找到最快的方法解决问题。

试想你在干草堆里寻找一根针。你立马就能分辨出自己看到的是一棵干草还是一根针，但是从哪里寻找却是一个很大的问题。你有很大的选择空间，但问题是：“怎样做出选择才能让我找到解法？”

非确定型图灵机的道理与之大同小异。它的问题是：“是否存在某种特定的指令，可以使我成功求解？”如果这样的指令存在，那么它就会惊呼“太好了！”然后遵照指令，在最快的时间内找到解法。如果这样的指令不存在，那么它就会唏嘘“太可惜了”，然后停止运转。

至于这类聪明的图灵机是如何判断出解题方法的，这一点在某种程度上讲还是个迷。对此，人们设想了两种情况。第一，答案已经摆在那里了。这就好比 you 有一块魔镜，它无所不知，每次都会告诉你：这是最好的选择。第二，可以采用某种平行或并行操作，也就是说，这类非确定型

图灵机所做的，其实就是同时运行所有可能的选择。

这些奇怪的思想是由图灵等业界先驱同时提出的，它们历经发展演进，为一个新的理论研究领域提供了肥沃的土壤，这个研究领域叫做可计算性理论（有时又称为递归论）。可计算性理论研究的是不可计算函数，研究人员致力于探索它们的不可计算性究竟达到了何种地步——或许有些函数比其他函数更难，尽管它们都不可计算。这些思想听起来或许有些异想天开，但是，有好些非常重要的研究成果可以转化为实际应用。其中有个例子就是莱斯定理（Rice's theorem）。莱斯定理的内容是，对于图灵机使用的特定语言，我们无法判定它是否具有非平凡性⁽¹⁸⁾。这句话放到编程语言上就是指，没有任何通用的方法可以判定关于语言的非平凡问题。简单说来，这意味着，要想写出能够完美调试其他程序的计算机程序，是不可能做到的事情。换言之，我们永远无法避免软件出现错误。

另一个重要的思想涉及到一个概念，称为字符串复杂性。在计算机科学中，字符串就是指由字母或数字组成的有限序列，比如“`abcdefghijklmnop7890`”。它可以代表任何事物，比如一个英语句子、一幅图像，甚至一首曲子。一个字符串的复杂度可以通过“柯尔莫

哥洛夫复杂性”（Kolmogorov complexity）来衡量。至于柯氏复杂性的界定依据是什么，想必读者朋友们已经猜到了——没错，那就是图灵机。在给定输入为 w 的情况下，如果图灵机 M 能够生成某个字符串，而且采用了最简洁、最紧凑、最高效的生成方法，那么该字符串的柯氏复杂性就是 M 与 w 的编码长度之和。也就是说，柯氏复杂性与字符串看起来有多复杂无关，而纯粹取决于生成字符串的难度。

分形⁽¹⁹⁾几何图形就是一个很好的例子。曼德勃罗集⁽²⁰⁾（Mandelbrot set）是一个复杂到无以复加的分形几何图形，不管你怎么放大，都可以看到无穷无尽的细节图案。它们多姿多彩，美不胜收，有的像海岸线，有的像有机分子结构，还有的像星罗棋布的河流。但是，纵使曼德勃罗集的图案千变万化，它的本质就是一个点集，这个点集出自一个非常简洁的方程式： $x_{t+1} = x_t^2 + c$ ，所以它的柯氏复杂性非常低，也就是说，我们只需要很少的信息就可以生成曼德勃罗集。

至于柯氏复杂性会产生什么效果，你只要会用计算机，就一定不会陌生，只不过你自己可能还没有意识到这一点。你可以在计算机上任选几个文件，用zip之类的压缩软件将它们压缩，这时你就会发现它们的大小缩水了，而且有些文件的缩水幅度比其他文件大很多。精密复杂的图片压

缩起来没有简单的图片缩水幅度大。这是因为解压缩程序需要足够的信息才能生成原始图片，按照定义，复杂的图片需要较多的信息才能生成。

至于文件可以压缩到多小的地步，这是有理论极限的。至于极限究竟何在，我们可以通过使用精巧的压缩程序，找到些许蛛丝马迹。但是，或许颇具讽刺意味的是，没有任何通用的函数可以计算任意字符串的柯氏复杂性。这样的函数是不可计算的。

就算我们算不出具体的极限值，也会希望文件的压缩幅度能够超越理论的极限，因此，有损压缩⁽²¹⁾软件应运而生。MP3就是一个典型的例子，它的压缩幅度超过了理论的极限，压缩过程中损失了一定的信息，因此，与未经压缩的原始CD文件相比，MP3文件播放起来有一定程度的失真。音乐文件在保存为压缩格式的过程中，只损失了人耳不易分辨的信息，但是如果将一首曲子的MP3版和CD版对照起来播放，你还是能听出其中的细微差别。同样的道理也适用于jpeg和tiff格式的图片。职业摄影师往往会将图片保存为未经压缩的tiff格式，因为保存为jpeg压缩格式会丢失一定的信息，从而使图片的质量受到些许影响。你要是仔细查看jpeg格式的图像，就会发现，有些细节部分似乎显得模糊不清。但是tiff和jpeg文件在大小上差别太大，所以大多数人还是会选择

jpeg文件。

理论前景

艾伦·图灵及其同时代的人为计算机科学的诞生奠定了理论基础，不仅如此，很多人认为，是他们开创了计算机科学本身。图灵的悲惨境遇和英年早逝并没有使他的成就埋没于历史的尘埃中，相反，他的思想生根发芽，成为了人类知识宝库中的参天巨树，昭示着计算机的力量和局限。

理论计算机科学可能有时候给人的感觉像是脑筋急转弯，但是它与人们的生活一直息息相关。随着计算机的迅速普及，图灵开创的可计算性理论和计算复杂度理论也变得日益重要。我们将在后面的章节中提到，互联网正成为沟通整个世界的信息网和关系网，在信息模式日益复杂的情况下，要想检索出自己所需要的信息，已变得非常困难。软件正变得日益精巧，这给软件的调试和兼容带来了很大的挑战。只要我们能更加透彻地理解计算的理论问题，我们在解决实际问题的过程中就会更加得心应手。“如今，我们需要处理的数据量极为庞大，解决理论问题是一件刻不容缓的事情，”赫希表示，“因此我们需要构建更多的计算模型，这样就能在相对较短的时间内处理更多的数据……或许这意味着，理论问题已

经变得空前重要。”

“P是否等于NP”这个难题的存在，也给很多数学家带来了不懈研究的强大动力。马克·赫布斯特对它的重要性坚信不疑。“在当今时代，理论计算机科学领域最重要的思想，就是时间复杂度等级的构建及其研究，它们是解决 $P=NP$ 问题的关键。 $P=NP$ 问题是数学和计算机科学领域最重要的一大未解之谜。”

图灵要是在世，或许已经为我们解开了 $P=NP$ 问题。不过，千禧年难题百万大奖依然有效， $P=NP$ 这道难关一旦被攻克，计算机的性能和效率必将提升一个台阶，届时，一场新的计算机革命必将势不可挡。

注释

[\[1\]](#) 一战后，随着英国国外煤炭危机的发生和国内经济的衰退，英国煤炭企业经营陷入了困境，煤矿资本家为了提高煤炭市场竞争力、降低经营成本，企图对矿工采取削减工资、增加工时的政策，引起了矿工的极大不满。当矿工与煤矿资本家之间的矛盾被激化时，随即导致了1926年这场声势浩大、席卷全英国的总罢工。

[\[2\]](#) 傅科摆（Foucault pendulum）指仅受引力和吊线张力作用而在惯性空间固定平面内运动的摆。为了证明地球在自转，法国物理学家傅科（1819—1868）于1851年做了一次成功的摆动实验，从而有力

地证明了地球是在自转，傅科摆由此而得名。

(3) 剑桥大学内最有名的学院之一，成立于1441年，由当时的英国国王亨利六世设立创建，因而得名“国王”学院。

(4) 中心极限定理是概率论中的一组定理。中心极限定理说明，大量相互独立的随机变量，其均值的分布以正态分布为极限。这组定理是数理统计学和误差分析的理论基础，指出了大量随机变量之和近似服从正态分布的条件。

(5) 詹姆斯·威尔金森，1919年-1986年，英国数学家和计算机学家，在数值分析领域具有杰出贡献，曾经协助图灵设计计算机Pilot ACE。

(6) 形式语言是按一定规律构成的句子或符号串的有限或无限的集合。

(7) 在数学领域里，形式证明是形式系统的产物，由一些公理与演绎规则组成。理论便是形式证明可能的最后一行结论。这几个步骤总和起来便是数学界通称的形式主义，但它更应该称做有限主义。

(8) 英国政府通信总部的情报机构。英国通信总部相当于国家安全局。

(9) 在密码学史中，恩尼格玛密码机（德语：Enigma，又译哑谜机）是一种用于加密与解密文件的密码机。确切地说，恩尼格玛是对二战时期纳粹德国使用的一系列相似的转子机械加解密机器的统称，它包括了许多不同的型号。

(10) 个体发生学（或称做形态发生）描述一个生命体从受精卵到成体的起源和发育。个体发生学在发育生物学、发展心理学和发展生理学中被研究。

(11) 罗杰·彭罗斯（1931年8月8日—），是英国

数学物理学家与牛津大学数学系W. W. Rouse Ball名誉教授。他在数学物理方面的工作拥有高度评价，特别是对广义相对论与宇宙学方面的贡献。

(12) 在计算机科学中，算法的时间复杂度是一个函数，它定量描述了该算法的运行时间。这是一个关于代表算法输入值的字符串长度的函数。

(13) 空间复杂度 (Space Complexity) 是对一个算法在运行过程中临时占用存储空间大小的量度。

(14) 所谓多项式时间 (Polynomial time)，就是指问题需要的时间 (复杂度) 与问题的规模之间是多项式关系。

(15) 指数时间，计算机算法术语。在计算复杂度理论中，指数时间指的是一个问题求解所需要的计算时间 $m(n)$ ，依输入资料的大小 n 而呈指数增长 (即输入资料的数量依线性成长，所花的时间将会以指数增长)。

(16) 在数学中，所谓病态的问题是指这样的数学问题：当它的输入数据 (如参数、初始值等) 有微小摄动时，会引起解的大扰动。相反的情形是良态数学问题。

(17) 在计算机程序或文本编辑中，硬编码是指将可变量用一个固定值来代替的方法。

(18) 数学中，术语“平凡” (平凡的) 经常用于结构非常简单的对象 (比如群或拓扑空间)。有时亦会用明显或乏趣这两个词代替。对非数学工作者来说，它们有时可能比其他更复杂的对象更难想象或理解。

(19) 分形图形一般都有自相似性，这就是说如果将分形图形的局部不断放大并进行观察，将发现精

细的结构，如果再放大，就会再度出现更精细的结构，可谓层出不穷，永无止境。

[\(20\)](#) 曼德勃罗集的提出者是伯努瓦·曼德勃罗（Benoit Mandelbrot）。他是波兰裔美国数学家、经济学家，分形理论的创始人。

[\(21\)](#) 所谓有损压缩是利用了人类对图像或声波中的某些频率成分不敏感的特性，允许压缩过程中损失一定的信息；虽然不能完全恢复原始数据，但是所损失的部分对理解原始图像的影响缩小，却换来了大得多的压缩比。

一次性的计算

计算机从问世至今，已经历了十亿次的更新换代。第一代计算机动辄耗资数百万美元，体型庞大到足以占满一座宽敞的大厅。如今的计算机灵巧质轻、物美价廉，它们的更新换代日新月异，被淘汰的计算机就像一次性糖纸一样，被我们用完即弃。

通用的设计和共同的语言使计算机能够互相交流，掌控我们的世界。它们遵守自己的规则，不受我们的约束。摩尔定律⁽¹⁾（Law of Moore）奠定了它们无所不在的地位。

但是，计算机在追求更快捷、更小巧的道路上能走多远？当它们的欲望和大小开始挑战物理定律的极限时，究竟会发生什么样的事情？它们会不会彻底转变成完全不同的事物？我们能否满足它们未来的需求？

在贝尔电话实验室⁽²⁾（Bell Telephone Laboratories）的食堂里，员工们正一边用餐，一边低声交谈。突然，一道尖利的嗓音划破了原本平静的氛围。“不，我对制造强有力的大脑根本没有兴趣，我感兴趣的只是平庸的大脑，比如美国电话电报公司（American Telephone & Telegraph Company，简称AT&T）总裁的大脑。”说话的人正是艾伦·图灵。

时值1943年初，图灵正在访问他赴美之行的最后一站——贝尔实验室。他之所以来到这里，

就是为了协助大西洋通信的语音加密工作（说白了就是给大西洋两岸传输的通话内容加密，这样敌人就无法监听）。不过，这次访问很快就因为另一个原因而变得收获颇丰。每天下午茶时间，图灵都会和实验室里的一位研究人员在食堂里长谈，这位研究人员名叫克劳德·香农（Claude Shannon）。两个人似乎都对计算机的问题非常热衷。图灵看问题主要从数学的视角出发，而香农的视角则完全不同。

克劳德·香农比图灵小四岁。他出生在美国密歇根湖畔的一个小镇——皮托斯基

（Petoskey），父亲是商人，母亲是盖洛德高中（Gay Lord High School）的校长。克劳德在邻镇盖洛德长大，后来进入母亲掌管的学校就读。他从小就对工程学和数学表现出浓厚的兴趣。很小的时候，他就喜欢捣鼓Erector公司生产的金属建筑拼装玩具，经常自制飞机模型和无线电遥控船，还在自家和半英里（约合805米）以外的朋友家之间建立了一个电报系统（为了建立这个系统，还专门利用了附近一家牧场周围的带刺铁丝网）。他在业余时间从事送报纸和发电报的工作，还给附近的一家百货公司修理收音机。他的儿时偶像是托马斯·爱迪生，后来他发现爱迪生其实是他的远房表亲——两个人都是殖民时期的领袖约翰·奥格登（John Ogden）的后裔。

香农虽然生性羞怯，但是对踩单轮车等平衡杂技非常痴迷。他的动手实干能力很强，经常把自己的数学思想运用到一些五花八门的实践当中，比如制造杂耍器材，发明一些稀奇古怪的单轮车等等。

到了20岁，这名性格内向、为人谦逊的美国小伙子从密歇根大学安娜堡分校（University of Michigan, Ann Arbor）顺利毕业，获得了电子工程和数学双学位。接着，他入职享誉世界的名校——麻省理工学院（MIT），担任研究助理，辅助操作微分分析仪。微分分析仪是该校工程学院院长万尼瓦尔·布什（Vannevar Bush）几年前制造的一个机械装置，由齿轮和螺丝钉组成，还有复杂的机电控制电路，每次计算新的方程式之前，都需要为期好几天的重新装配。由于香农一向擅长修理器械，操作微分分析仪的任务自然就交给了他。

一年后，香农进入贝尔电话实验室实习，学习操作自动电话交换机^{[\(3\)](#)}。实习期间，他意识到了一件重要的事情。年轻的香农发现，有两个看似截然不同的事物其实具有共同的本质。

逻辑思维

克劳德·香农回到麻省理工学院，发展他的理论新思想。那时候，他还没有和图灵见过面，图

灵当时还是邱奇的学生，正在250英里（约合402公里）以外的普林斯顿大学攻读博士。两人共进午餐是六年以后的事情。

香农已经知道，数学上有一种逻辑代数系统，叫做布尔逻辑，它得名于英国数学家乔治·布尔（George Boole）。在布尔逻辑中，任何逻辑表达式的计算结果都不是数值，而

是“真”、“假”这两种真值^[4]。你只需要使用逻辑运算符“与”、“或”、“非”，就可以表达任何你想表达的逻辑语句。这个逻辑语句可以是一个英文句子，比如“在下雨且阴天或无风的时候，我会带伞。”

通过考察所有的可能性，我们可以把这个句子用布尔逻辑表达出来。我们把带不带伞的决定称为“决定Q”，如果Q为真，就带伞；如果Q为假，就把伞留在家里。影响Q的三个因素分别用三个字母来代替：A表示“下雨”，B表示“阴天”，C表示“无风”。因此，无论A、B、C取什么样的可能值，Q（带不带伞的决定）都会有一个真值与之对应。如果我们把所有可能值列举出来，就会得到一个真值表^[5]：

A (下雨)	B (阴天)	C (无风)	Q (带伞)
假	假	假	假
真	假	假	假
假	真	假	假
真	真	假	真
假	假	真	假
真	假	真	真
假	真	真	假
真	真	真	真

根据这张表，我们可以写出一个对应的布尔表达式。从表中可以看到，Q只在三种情况下为真：第一，A为真，B为假，C为真；第二，A为真，B为真，C为假；第三，A为真，B为真，C为真。我们可以用类似于布尔代数的写法将Q与A、B、C的关系表述出来：

$$Q = (A \text{ 与 非} B \text{ 与 } C) \text{ 或 } (A \text{ 与 } B \text{ 与 非 } C) \text{ 或 } (A \text{ 与 } B \text{ 与 } C)$$

这个句子还可以用笨一点的方法表述出来：“在下雨、不阴天且无风，或下雨、阴天且有风，或下雨、阴天、且无风的时候，我会打伞。”不过，布尔代数理论当中，有许多实用的规则可以帮助我们逐步简化这个表达式，以下是精简后的表达式：

$$Q = A \text{ 与 } (B \text{ 或 } C)$$

你要是仔细查看这个表达式，就会发现，它可以将上述句子的内容完美地表现出来。由此可见，布尔逻辑可以让我们描述和操纵逻辑表达式，这就和我们通过数学函数来操纵数字是一样的道理（正如上一章所讲的那样，所有的数学问题都可以归结为逻辑问题）。

香农取得的突破在于，他注意到，逻辑和开关电路⁽⁶⁾具有共同的本质。他借鉴了布尔逻辑，并运用它来定义带有机电式继电器（电气开关）的电路。此外，他还将开关的连接方式改写成了逻辑表达式，详见图2。

其中的原理非常简单。如果开关处于“接通”状态，那么在布尔逻辑当中就相当于真值为“真”；如果开关处于“断开”状态，那么在布尔逻辑当中就相当于真值为“假”。如果你想设计一个电路，让它像逻辑运算符“与”一样运转，那就可以把开关串联（也就是将两个开关头尾相连，逐个顺次连接）。这样一来，电路只有在两个开关都接通的情况下才会接通。这和逻辑运算符“与”（有时候写成一个“+”号）的特点相符，因为逻辑表达式“X与Y”只有在X和Y都为真的情况下才为真。同理，如果你想设计一个电路，让它像逻辑运算符“或”一样运转，那就可以把开关并联（也就是将两个开关头头相连，并列连接在两点之间）。这样一来，只要接通其中任意一个

开关，就可以接通整个电路；如果两个开关都断开，则整个电路也断开。这和逻辑运算符“或”（有时候写成一个“.”号）的特点相符，因为逻辑表达式“X或Y”为真的情况有三种：X为真、Y为真或二者都为真。香农的理论表明，整个电路都可以用布尔逻辑表述出来，只要巧妙地运用逻辑表达式，就可以简化和改善电路设计。



图2 香农的与门、或门逻辑电路

也就是在这个时候，人们开始着手制造世界上第一台电子计算机。由此可见，只要能够运用逻辑数学表达式设计出简洁而高效的电路，就能创造巨大的实用价值。由于所有的数学问题都可以归结为逻辑问题，而逻辑问题又可以通过电气开关表现出来，香农的理论表明，人们可以设计专门的电机，用来计算任何可计算的数学函数。

香农决定将他的思想写成研究报告发表出来，研究报告的题目是“继电器和开关电路的符号分析”（A Symbolic Analysis of Relay and Switching Circuits）。这项成果给计算机科学领域带来了重大突破。香农很快以此为基础，完成了他在麻省理工学院的硕士论文。这篇论文受到了广泛的赞誉，人们说它“或许是本世纪最重要、

也最有名的硕士论文”。香农也因此获得了美国土木工程师协会（American Society of Civil Engineers，简称ASCE）颁发的“阿尔弗雷德·诺贝尔奖”（Alfred Noble Prize）。尽管如此，香农依然保持着一贯的谦逊作风，只是平静地说：“我只是想知道不同的事物的联系而已。”

24岁时，香农写了一篇博士论文，从代数学的角度描述遗传学和进化论。毕业后，他作为国家研究员（National Research Fellow），在声名远播的普林斯顿高等研究院⁽⁷⁾（Institute for Advanced Study，简称IAS）工作了一年。在那里，他接触到了—些世界上最顶尖的人才，其中包括赫尔曼·外尔⁽⁸⁾（Hermann Weyl）、阿尔伯特·爱因斯坦、库尔特·哥德尔和约翰·冯·诺依曼（John von Neumann）。1941年，香农进入贝尔电话实验室，继续扩充自己的理论思想。

两年后，在贝尔实验室的食堂，图灵在与香农交谈的过程中大为振奋——他的声音不由自主地越来越大，引起周围的人纷纷侧目。香农的话让他看到了希望曙光——图灵机或许真的可以变成现实！临走前，图灵买了一本电路入门书，把它带到回国的船上，在危险四伏的海上航程中如饥似渴地阅读起来。

制造大脑

事有凑巧，1943年，当图灵回到英国后，一位名叫约翰·冯·诺依曼的数学家也从美国漂洋过海，来到了英国。冯·诺依曼当时是普林斯顿高等研究院最年轻的成员（他是最先被研究院聘为教授的五人之一，同时入院的还有爱因斯坦），他人脉很广，也认识图灵，因为在1938年，图灵博士毕业，成为研究助理后，他曾要求图灵留在普林斯顿。但图灵拒绝了这个工作机会，回到了英国剑桥。冯·诺依曼也见过香农。那是在1940年，香农还在高级研究所担任研究员。

在英国，冯·诺依曼被布置了一项任务：用数学方法预测德军在英军护航⁽⁹⁾线路上布置磁性水雷的模式。这件事情他不费吹灰之力就可以办到。访英期间，冯·诺依曼还在巴斯（英格兰埃文斯郡东部小城）的航海年历局（Nautical Almanac Office）设计了他的第一个数值计算程序，安装在收银机NCR 3000（一台电动加法机）上。（在此期间，他可能拜访过图灵，但是没有确凿的证据显示两人见了面。）他给家人的信中写道：“……我还对计算方法产生了近乎疯狂的兴趣。”后来，这份兴趣带来了他自己都未曾想到的结果。

约翰尼·冯·诺依曼（朋友们都这么叫他）从小到大大一直是个杰出的天才。他1903年出生于匈牙利布达佩斯，匈牙利语原名叫做亚诺什·诺依曼

（János Neumann），小时候就表现出惊人的天赋。父亲经常带着小扬希（Jancsi，亚诺什的昵称）去参加派对，因为他可以在几秒钟之内把电话簿的一整页内容完整地记下来，然后准确无误地将上面的姓名、住址和电话号码默写出来。冯·诺依曼的父亲是个顶尖的银行家（他在1913年买了一个爵位，诺伊曼家族由此晋身为贵族，姓名中带上了一个“冯”字）。亚诺什从小跟着两名女家庭教师学习法语和德语，1911年入读声望卓著的路德教会文理中学（Lutheran Gymnasium）。在那里，他优异的数学才能得到了进一步的培养。

冯·诺依曼在校期间表现出了惊人的数学天赋，但他父亲希望他能学一个将来好找工作的专业，于是冯·诺依曼选择了在柏林大学

（University of Berlin）攻读化学专业。（他父亲作为银行家，一直希望儿子能学商业，不过，学化学是两个人各让一步的结果。）从1921年到1923年，他在柏林大学就读，后来进入苏黎世大学（University of Zurich）。在校期间，冯·诺依曼利用业余时间修读数学课程，他的天赋很快受到了重视。一名教过他的数学家后来承认：“约翰尼是我害怕过的唯一一名学生。如果我在一堂讲座上公布了一道未解的难题，他通常会在讲座一结束就找到我，给我一张字条，上面草草地写下了完整的解法。”1926年，冯·诺依曼拿到了化学

工程学位，同一年又在布达佩斯大学拿到了数学博士学位，充分说明了他真正感兴趣的领域其实是数学。冯·诺依曼从1926年到1931年相继在好几所高校担任讲师，1931年被高等研究院聘为教授。他一方面潜心研究量子力学，另一方面解决了不少数学问题，成为了数学领域举足轻重的奠基人。到了这个时候，他已经被其他经验丰富的数学家视为年轻有为的天才，俨然被明星的光环笼罩。他开始定期造访附近的普林斯顿大学，图灵当时还是邱奇的学生，正在普林斯顿大学开展博士研究。

20世纪30年代，人们越来越热衷于发明自动计算机。几个世纪以来，机械式计算机除了齿轮和螺丝钉以外，什么部件也没有。冯·诺依曼对这些设备深为着迷，他尤其痴迷于19世纪20年代查尔斯·巴贝奇（Charles Babbage）发明的一台机械式计算机，它与如今的现代计算机在设计上有许多共通之处。随着继电器这种电控开关的问世，利用电动设备进行计算已经成为了可能。不少工程师发明了早期计算机。其中最早的机型之一采用了柏林工程师康拉德·楚泽⁽¹⁰⁾（Konrad Zuse）发明的继电器。这台巨型机器称为Z3，由于缺少条件分支⁽¹¹⁾，它的功能受到了一定的局限。也就是说，它不能根据不同的计算结果执行不同的操作，必须在程序中不停地执行相同的计

算。它的运行速度也很慢，因为继电器采用了移动部件接通和断开电流。

运行速度更快的计算机都采用了没有移动部件的电子管（真空管）。20世纪30年代，美国爱荷华州立大学（Iowa State University）的约翰·阿塔纳索夫（John Atanasoff）花了几年的时间发明了一台电子计算机，用来解线性代数方程。这台计算机由他和学生克利福德·贝里（Clifford Berry）共同制造，称为“阿塔纳索夫-克利福德贝里计算机”（Atanasoff-Berry Computer，简称ABC）。它体型较小，性能不甚可靠，内含大约300个电子管。与此同时，英国工程师托马斯·弗劳尔斯（Thomas Flowers）于1934年在继电器的基础上发明了独创的开关系统，并于30年代末投入使用。开关系统采用了3000多个电子管，用于英国电话交换机和简单的数据处理。

1943年，图灵回国后，鼓动马克斯·纽曼接近弗劳尔斯，请他来布莱切利公园，帮忙改进他们在继电器的基础上设计出来的破译机。1943年底，弗劳尔斯发明了巨像I（Colossus I）——一台内含1600个电子管的电子计算机。巨像I后来一共制造了十台，每一台都包含2400个电子管，但它们不是通用机器，必须插入不同的电缆重新编程。

这时候，冯·诺依曼已结束英国之行，回到了

美国。由于他名气很大，美国国防部邀请他参与曼哈顿计划（Manhattan Project，美国政府制造第一颗原子弹的计划），负责设计原子弹的爆炸外护层（explosive outer jacket）。要完成这项工作，必须进行大量复杂的数学计算。冯·诺依曼意识到，他需要一台全新的计算机，其性能必须远远超越当前所有的计算机。

一次偶然的相遇注定将改变一切。1944年夏，冯·诺依曼在马里兰州的火车站遇到了曾任数学教授的赫尔曼·戈德斯坦中尉（Herman Goldstine）。戈德斯坦当时是宾夕法尼亚大学（University of Pennsylvania）摩尔电气工程学院（Moore School of Electrical Engineering）的军方联络人。在摩尔电气工程学院，一台令人惊叹的全新计算机正在紧锣密鼓的研制当中。时至今日，戈德斯坦依然对当年与冯·诺依曼交谈的情形记忆犹新：“话题很快就转移到了我的工作上。我说我正在关注一项开发任务，任务的目的是制造一台每秒钟可以运算333个乘法的电子计算机。这句话才说完，整个谈话的气氛一下子就变了，原本我们还是比较轻松、随意地聊聊天、打打趣，突然变得像是在做数学博士学位的口头答辩。”

冯·诺依曼马上做出安排，拜访了项目的设计师：约翰·莫奇利（John Mauchly）和约翰·埃克特

（John Presper Eckert）。他们设计的是世界上第一台通用计算机：电子数字积分计算机（The Electronic Numerical Integrator and Computer）——简称“埃尼阿克”（ENIAC）（如图3）。这个庞然大物尺寸为 $8\times 3\times 100$ 英寸（约合 $2.4\text{米}\times 0.9\text{米}\times 30\text{米}$ ），重约三十吨，包含17,000多个电子管和1500多个继电器，运算速度比以往的任何计算机都快。冯·诺依曼很快就开始定期造访摩尔电气工程学院，并受邀参与了埃尼阿克的设计项目。



图3 电子数字积分计算机（埃尼阿克）

从左到右：霍默·斯彭斯教授（Homer Spence）；首席工程师约翰·埃克特；顾问工程师约翰·莫奇利博士；贝蒂·让·詹宁斯（Betty Jean Jennings）；弹道实验室⁽¹²⁾——宾夕法尼亚大学联络官赫尔曼·戈德斯坦（已升为上尉）；露丝·里克特曼（Ruth Lichterman），1946年摄

埃尼阿克的研制工作进展缓慢，因此，在研制工作完成之前，军方又布置了一个任务，要求再建造一台更快的计算机。埃尼阿克的后继者称为电子离散变量自动计算机（Electronic Discrete Variable Automatic Computer）——简称“埃德瓦克”（EDVAC），约翰尼·冯·诺依曼成为了设计团队的一员。埃德瓦克的设计持续了好几个月，涉及到很多新思想和新技术。挑战主要来自两大方面，一是保存数据的存储器，需要解决的问题是，数据能否存储在某种形式的雷达甚至电视显像管里？二是指令系统，这里需要考虑的是，有哪些功能是计算机应该具备的？1945年6月，冯·诺依曼撰写了一篇文章，对摩尔学院项目团队的设计理念进行了总结。这只是一份初稿，署名作者只有他一个人，但是文章的终稿版从未出炉。戈德斯坦鼓励将初稿的内容公诸于世，因此，埃德瓦克的设计思想很快在全世界研究人员和工程师中传播开来。文章标题为“关于埃德瓦克的报

告初稿”(First Draft of a Report on the EDVAC)。这是第一份系统描述计算机制造方法的公开出版物，具有划时代的革新意义。

数字大脑的解剖

尽管“关于埃德瓦克的报告初稿”描述的是工程学和数学领域的研究成果，但它的遣词造句通俗而又浅显，几乎人人都能看懂。报告是冯·诺依曼1945年在火车上手写的，以下是部分摘录：

我们只要分析一下这台构想中的设备应该具备哪些功能，就不难看出，该设备的核心部件应该分为几个大类。

第一：由于该设备本质上是一台计算机，最起码要能够迅速地执行基本的算术运算，其中包括加法、乘法和除法。由此可见，该设备应该包含专门执行此类运算的部件……也就是说，中央算术器（central arithmetic part）或许是一个必不可缺的部件，它也构成了该设备的第一大部件，我们将其简称为CA。

第二：该设备需要对不同的运算操作进行适当的排序，这一过程称为逻辑控制。中央控制器（central control organ）可以对设备进行高效的逻辑控制……它构成了该设备的第二大部件，我们将其简称为CC。

第三：任何设备要想进行长时间的复杂操作

（尤其是计算操作），都需要容量相当可观的存储器（memory）.....它构成了该设备的第三大部件，我们将其简称为M。

.....CA、CC、M这三大部件就相当于人体神经系统中的联络神经元⁽¹³⁾。至于哪些部件对应于人的感觉神经元⁽¹⁴⁾（又称传入神经元）和运动神经元⁽¹⁵⁾（又称传出神经元），是接下来要讨论的问题。这些部件是该设备的输入装置和输出装置。

值得注意的是，这个设计方案包含五大逻辑元件：第一是中央算术器，负责执行所有的运算操作；第二是中央控制器，它决定机器的下一步动作；第三是存储器，用于保存程序及程序输出的结果；第四是输入设备，比如键盘；第五是输出设备，比如打印机。

为了方便大家更直观地理解埃德瓦克的工作机制，我们可以把上述五大逻辑元件设想成五个好友在互相交谈。假设五个人的名字分别为：康妮、阿尔伯特、玛丽、伊恩和奥利弗。康妮很有领导风范，喜欢发号施令。阿尔伯特有点喜欢在代码本上钻牛角尖。尽管代码本上可能没几个代码，但他总是会把每个指令的意义都弄明白，然后将结果告诉其他几个朋友。他还可以短暂地记住一些事情。玛丽拥有卓越的文件编排系统。无论让她查找什么信息，她都能很快给出结果。伊

恩和奥利弗是那种嘴巴停不下来的人。伊恩老是让玛丽记东西；奥利弗则喜欢到处找人侃大山。

在谈话过程中，五个人的角色始终如一。至于他们会谈什么，我给大家举一个例子：

康妮：“玛丽，伊恩有件事情想跟你说。”

伊恩：“玛丽，请帮我在你的文件柜里存七个数字，这七个数字分别是：100、010、001、101、110、110、000。”

玛丽：“好的，知道了。”

康妮：“阿尔伯特，你倒是也做点事情呀！”

阿尔伯特：“好吧，马上就做。玛丽，你能不能告诉我，刚才伊恩给你的第一个数字是什么？”

玛丽：“第一个数字是100。”

阿尔伯特：“嗯，有意思，我的代码本上说，100指的是，将后面两个数字相加。”

康妮：“玛丽，把后面两个数字给阿尔伯特。”

玛丽：“阿尔伯特，后面两个数字是010和001。”

阿尔伯特：“收到。嗯，从代码本上看，010和001相加的和是011。这个结果我先记着。”

康妮：“阿尔伯特，别在那儿偷懒，接着干活！”

阿尔伯特：“玛丽，你能不能告诉我，下一个数字是什么？”

玛丽：“是101。”

阿尔伯特：“我知道了，从代码本上看，这个数

字指的是，将上一个运算结果存储到下一个数字所指的位置上。”

康妮：“玛丽，把下一个数字给阿尔伯特。”

玛丽：“下一个数字是110，阿尔伯特。”

阿尔伯特：“感谢二位。也就是说，我得把数字011存储在文件110中。”

康妮：“玛丽，阿尔伯特想把数字011存储在文件110中。”

玛丽：“明白了，这就去办。”

康妮：“阿尔伯特，再做点事情！”

阿尔伯特：“好吧，康妮，我做就是了。玛丽，下一个数字是什么？”

玛丽：“下一个数字还是110。”

阿尔伯特：“唔……从代码本上看，110指的是，将上一个运算结果告诉奥利弗。”

康妮：“奥利弗，阿尔伯特有个数字要告诉你，是011。”

奥利弗：“真的吗？谢谢你，阿尔伯特。”

康妮：“阿尔伯特，接着干活！”

阿尔伯特：“好吧，玛丽，下一个数字是什么？”

玛丽：“000。”

阿尔伯特：“唔，代码本上说，000代表停机。”

康妮：“好了，大家都停下来！”

这时候，所有的交谈通常都会戛然而止。

康妮、玛丽、阿尔伯特、伊恩和奥利弗分别对应于埃德瓦克的五大逻辑元件。康妮代表的是中央控制器——她根据阿尔伯特对代码的解读，告诉所有人接下来该怎么做。

阿尔伯特对应的是算术逻辑单元（arithmetic and logic unit，简称ALU）和寄存器^{[\(16\)](#)}

（register），用冯·诺依曼的说法就是中央算术器。这个电子装置接收外界输入的电信号，并据此输出各种各样的电信号（即运算结果）。ALU的运作模式固定不变，受到布尔逻辑电路的支配。它设有临时存储区域，称为寄存器，寄存器的运作方式和存储器一样（但是访问速度要快很多），用于保存ALU和存储器输出的结果。

玛丽代表的是存储器。信息以二进制编码1和0的形式存储下来，二进制编码1和0则分别由“高”、“低”两种电压转换而来。在现代计算机中，这种类型的存储器通常被称为随机存取存储器（random access memory，简称RAM）。RAM这个缩写虽然看似神秘，但它的作用其实很简单。用通俗的话来讲，它可以帮助我们直接调取任何零散的信息，不需要像磁带或纸带那样，在庞杂的信息当中大海捞针。

伊恩对应的是输入电路，它为键盘等输入设备提供接口^{[\(17\)](#)}；奥利弗对应的则是输出电路，

它为显示器、打印机等输出设备提供接口。

上文的“谈话”展示的就是一个计算过程。由输入电路（伊恩）提供、并保存在存储器（玛丽）中的七个数字构成了一个程序，即一系列指令和数据的组合。由此开始了“提取-执行”的循环。处理器（康妮和阿尔伯特）依次询问每个数字的大小，将指令解码并加以执行。阿尔伯特负责执行，康妮负责指挥。执行的结果可以有很多种，比如在存储器中保存和读取数字、将计算结果发送到显示器等外围输出设备……如此等等，不一而足。只有当处理器接收到停机指令时，提取-执行的循环才会中止。

冯·诺依曼有很多设计理念（比如采用中央存储器，以保存数据和指令）明显受到了图灵机的影响。冯·诺依曼的同事、洛斯阿拉莫斯国家实验室[\(18\)](#)（Los Alamos National Laboratory）的物理学家斯坦利·弗兰克尔（Stanley Frankel）深知图灵的工作在当时的重要性。“我知道，大约在1943-1944年这段时间，冯·诺依曼非常重视图灵在1936年写的论文……他让我看看这篇论文，于是我好好研究了一下。”

冯·诺依曼也很重视香农的布尔逻辑，深知这一理论的重要价值。他在报告中写道：

这台机器的一大重要方面在于，它依据的不是算术原理，而是逻辑原理。非真即假的逻辑系

统从本质上讲就是一个二进制的系统。

冯·诺依曼的报告初稿势必影响整个世界。手稿的内容被歌德斯坦公诸于世并广泛传播后，人人都开始效仿这个高效而杰出的设计方案。冯·诺依曼结构由此得名，并成为了日后几乎所有计算机的经典设计样板。它包含五大权力集中、却又各司其职的逻辑元件。或许在当今时代，人们已不再将这些元件比作“器官”或“神经元”，但几乎每一台计算机的背后都可以看到它们的身影。

猝不及防的结束

直到1951年，埃德瓦克才被制造出来。尽管它比埃尼阿克要小，但还是安装了6000个电子管（真空管），功率高达56千瓦，需要工业空调才能保持降温。

没有人赶在战争结束之前成功造出可以使用的通用计算机。事实上，冯·诺依曼和最初的设计师莫奇利和埃克特之间发生了些许不快，因为莫奇利和埃克特对他们的理念不受重视而忿忿不平，他们想将自己的设计方案申请专利，投入商业使用。冯·诺依曼（如图4）放弃了埃德瓦克，转而决定在高等研究院制造一台截然不同的计算机。受到此项工作的激发，很多其他计算机被相继设计出来。

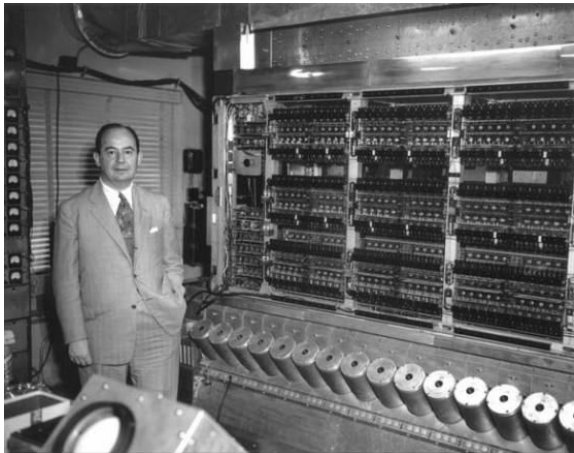


图4 约翰·冯·诺依曼和高等研究院（IAS）的第一台计算机，1951年摄

布鲁斯·戴默是加利福尼亚州圣克鲁斯山Digibarn计算机博物馆的馆长，他对计算机科学的早期发展史颇有研究。“约翰·冯·诺依曼在高等研究院开展的电子计算机项目（**electronic computer project**，简称ECP）比此前所有的项目都更加完善，”他表示，“整个系统运转速度极快，可以在不出现任何错误的情况下，长时间运行数据输入量和输出量都相当可观的程序（比如氢弹试验数据计算、天气预报、人工生命模

拟)。ECP的成果和设计方案得到了广泛共享。IBM当年之所以能推出商业计算机(700系列),就是因为合作参与ECP的过程中学到了关键技术。我们现在生活在冯·诺依曼构建的世界。现代的计算机系统虽然日新月异,但万变不离其宗,大框架都是冯·诺依曼1952年在普林斯顿构建起来的。”

但是,计算机领域的先驱并不只有冯·诺依曼一人。如今,有些人认为,“冯·诺依曼结构”这个提法本身就不够准确,因为做出贡献的还有其他人,比如莫里斯·威尔克斯。早在1949年,威尔克斯就在剑桥大学展示了第一台具有实用价值的存储程序计算机,其主要特点是采用了电子管作为基本电子元器件。尽管冯·诺依曼的报告和莫奇利、埃克特在摩尔学院的讲座启发了威尔克斯的设计灵感,但是这台计算机的问世时间比冯·诺依曼的IAS计算机(以及埃德瓦克)早了整整两年,而且威尔克斯团队的研究人员后来也加入到冯·诺依曼的团队当中,协助他们的设计工作。威尔克斯本人还提出了许多重要的设计思想,比如微代码⁽¹⁹⁾。

很快,全世界的工程师纷纷开始发挥聪明才智,为各大高校和大型企业制造他们自己的巨型计算机。这些机器无一例外都是庞然大物,极为费电,内含成千上万个电子管,且大多采用穿孔

卡片输入数据和指令。但它们造价高昂，故障频发，为了提高运算能力，不得不安装更多的电子管，这就意味着它们还得向体积更大、功率更高、性能更差的方向发展。

开发新技术的工作迫在眉睫，事实上，有一项新技术不久前已经问世。冯·诺依曼写下关于埃德瓦克的报告后，才过两年，贝尔实验室就发明了晶体管。从理论层面讲，晶体管能做到的事情和电子管完全相同，但从实际角度讲，晶体管具有无可比拟的优越性，不仅速度快了好几倍，而且耗电量和体积也小了很多。

1953年，世界上第一台晶体管计算机由曼切斯特大学研究生迪克·格里姆斯戴尔（Dick Grimsdale）制造完成。一石激起千层浪，此后15年里，世界各地涌现了一百多台晶体管计算机。但是，第一台晶体管计算机才问世五年，电子工程师就已经面临一个严峻的问题——“数字的暴政”（tyranny of numbers）：为了提高运算能力，就必须安装更多的晶体管，但是一旦安装更多的晶体管，制造和维护成本就会增大，因为每个部件都需要手工焊接到电路板上。

这个问题并没有让电子工程师困扰多久。1958年，德州仪器^{[\[20\]](#)}（Texas Instruments）公司的新职员杰克·基尔比（Jack Kilby）产生了一个革命性的想法：为什么不同时制造所有的元件

呢？只要蚀刻锗晶片，就可以将电路中所有的元件都集成到一小块芯片上。几个月后，到了1959年，另一位研究人员——罗伯特·诺伊斯（Robert Noyce）也独自想出了同样的主意，只不过他使用的不是锗晶片，而是硅晶片。集成电路（integrated circuit，简称IC），也就是硅芯片就此诞生。

这个发明对于冯·诺依曼来说，来得太迟了。他一直在学术上披荆斩棘，忙得不可开交，为量子物理、博弈论、核物理学、流体动力学和计算机科学领域做出了巨大的贡献。他持有强硬的政治观点——自始至终大力倡导使用核武器，实行先发制人的战略。此外，他享受着丰富而充实的社交生活——约翰尼·冯·诺依曼和第二任妻子举办的派对一时之间成为了传奇。然而好景不长，1955年，冯·诺依曼被查出患有骨癌。后来，他的同事尤金·维格纳（Eugene Wigner）回忆起了他当年遭受的痛苦，“冯·诺依曼知道自己换了绝症之后，理智迫使他承认，再过不久，他就不会在这个世界上存在，也无法再思考了……我们都很心痛，看着他和命运搏斗，直到失去所有希望，一天天地消沉下去。他知道自己反抗不了命运，但他接受不了。”

另一名同事爱德华·特勒（Edward Teller）也回忆起了昔日好友的最后时光：“我觉得，对冯·

诺依曼来说，被剥夺了思考的权利比死还难受。在这一点上，他比谁都痛苦。”

1957年，约翰·冯·诺依曼去世，时值图灵去世三年后，硅芯片问世两年后。

摩尔定律

发明硅芯片后，罗伯特·诺伊斯继续和同事戈登·摩尔（Gordon Moore）在飞兆半导体公司

（Fairchild Semiconductor Corporation）研究集成电路技术，直到1968年，两人成立了一家新公司，即著名的英特尔。他们和全世界的业内先驱一道，共同掀起了电子学领域的技术革新。早期的集成电路还只有几百个、乃至上千个晶体管，但制造工艺的稳步改进使单个芯片上可以容纳的晶体管数目越来越多。（20世纪60年代早期，集成电路技术之所以迅猛发展，很大程度上是因为美国导弹计划和阿波罗太空计划的推动。）

集成电路的日益复杂使戈登·摩尔在1965年做出了一个预测。当时，他注意到，从1958年集成电路问世到1965年，单个芯片上的晶体管数量每年都翻了一倍。于是，他预测，这个趋势至少还会持续十年。后来，他修正了自己的观点，认为单个芯片上的晶体管数量每两年就会翻一倍。

1970年，加州理工学院（California Institute of Technology简称Caltech）教授卡弗·米德（Carver

Mead) 发明了专门的术语，将这个预测称为“摩尔定律”。

令人惊叹的是，这个“定律”似乎一直都很准。单个芯片上可容纳的晶体管数量在20世纪70年代中期为一万个，在1986年达到了一百万个，在2005年则为十亿个。尽管经常有人提出，摩尔定律很快就会失效，因为晶体管的尺寸越做越小，已经快要达到物理定律的极限，不过到目前为止，这项卓越的技术依然保持着强劲的发展势头。

计算机技术一向与电子学的前沿技术联系紧密，因此从20世纪60年代开始，硅芯片技术的迅猛发展也带动了计算机的更新换代。摩尔在英特尔的同事大卫·豪斯 (David House) 认为，从摩尔定律可以推断出，计算机的性能每隔18个月就会翻一倍。他说的基本没错——这些年来，计算机的性能大约每隔20个月就翻了一倍。摩尔后来开玩笑说：“18个月是豪斯说的，不关我的事。”

加州大学伯克利分校 (University of California Berkeley) 教授戴维·帕特森 (Dave Patterson) 是计算机体系结构⁽²¹⁾ (computer architecture) 专家，编写了该领域的大多数主要教材，曾担任计算机协会⁽²²⁾ (Association for Computing Machinery) 会长。他还在计算机科学的发展史上写下了浓墨重彩的一笔。不仅他本人

协助发明了独立磁盘冗余阵列⁽²³⁾（Redundant Arrays of Inexpensive Disks，简称RAID），他的博士生导师杰拉尔德·埃斯特林（Gerald Estrin）还曾在1950年-1956年期间与冯·诺依曼在高等研究院共事。事实上，在20世纪50年代，冯·诺依曼本来想与埃斯特林一起进入加州大学洛杉矶分校（UCLA），可惜最终未能成行。

帕特森对摩尔做出的预测大唱赞歌。“摩尔定律是计算机科学领域的一大重要思想。摩尔提出这个定律以后，又写了一篇论文，预测如果摩尔定律为真，将出现什么情况。他的预测都应验了。在那个只有银行或高校买得起计算机的时代，他预测，个人计算机和车载计算机的问世只是时间问题。摩尔定律本来就已经是个大胆的预测，他把这个预测可能带来的影响都想到了，真是太了不起了。”

“正因为有了摩尔定律，”帕特森继续说道，“正因为单个芯片上可容纳的晶体管数量每隔一、两年就会翻一倍，现在的情况与上个世纪50年代相比，不知道好了多少倍。我们在提升速度的同时降低了成本。硬件领域的革新令人惊叹。正因为如此，计算技术才会变得如此普及；正因为如此，世界上半数以上的人都拥有了手机（我把手机看做带有无线通信系统的计算机）是摩尔定律让这一切成为了现实——它使计算机速

度更快，售价更低。集成电路的问世和晶体管数量的激增创造了奇迹。这些都让人叹为观止。”

正因为电子学正向着微型化的方向大幅迈进，我们的计算机每年都在变得更小巧、更便宜、更强大。早期的计算机都是庞然大物，很多情况下只能靠远程终端控制（一台大型计算机就需要很多个键盘和显示屏）。很快，小型计算机和个人计算机（台式机）相继问世。随着单一芯片上集成的元件继续增多，计算机的尺寸进一步缩水，由此产生了便携式个人计算机，也就是笔记本。后来又出现了更迷你的上网本、平板计算机和掌上设备，比如智能手机。计算机已变得小巧、廉价，足以在儿童玩具上实现复杂的功能，甚至在贺卡中嵌入音乐，而且由于它的更新换代速度太快，被淘汰的设备无需多想就可以直接丢弃。本已小巧玲珑的计算机还会向着更小巧、更便宜的方向迈进，这种发展趋势还没有任何减缓的迹象。

虽然有了摩尔定律这一强大利器，科学家并没有安于现状，不再绞尽脑汁寻找提升计算速度的新方法。事实上，计算机领域的创新远未止步。尽管所有计算机都具备冯·诺依曼当年在报告中提到的逻辑元件，但它们远非千篇一律。为了提升运算速度或效率，各种高明的优化设计方案层出不穷。举个例子，自从计算机问世以来，工

工程师就一直面临着两难的问题：要想扩大存储量，就必须牺牲速度。在存储量小的情况下，速度可以很快，但存储量一旦扩大，速度往往就会受到拖累。我们在日常生活中也经历过类似的事情——假如你有一张简短的历史购物单，要想在上面寻找某样东西，是一件轻而易举的事情，不需要花什么时间；但是，如果你写了二十年的日记，想在这些日记当中寻找某一天购买过的某样商品，那么寻找起来就会困难很多，花的时间也会很长。因此，回到计算机的问题上，要想兼顾速度和存储量，可以结合采用多种不同类型的存储器。如今的计算机处理器通常有如下配置：少量超快内部存储器⁽²⁴⁾（称为寄存器）；一个内部高速缓冲存储器⁽²⁵⁾（cache，简称高速缓存）（或许还会加装一个容量稍大、但速度稍慢的高速缓存）；一个比高速缓存更慢，但却更大的外部存储器；一个比外部存储器还慢，但却大很多的硬盘；此外或许还会加装一个比硬盘还慢，但却更大的备份存储器（磁带或硬盘）。只要在时机把握得当的情况下，将数据和指令从慢速存储器转移到快速存储器，计算机就能迅速调取信息，从而运行得更快。（这就好比你在写日记的时候，将重要的信息提取出来，写在一张纸上，这样日后查找起来就会更方便。）

现代计算机处理器还有一种典型的优化设计

方案，称为流水线（**pipelining**，又称管线）。其具体执行过程非常类似于工厂中的流水线。下面我给大家具体解释一下。工厂制造一辆车可能需要一整天，但是几乎每时每刻都有新车出厂。这是因为采用了流水线操作，生产汽车的流程被分为许多道工序，所有工序并行操作，不同的车辆同时进入不同的工序。比方说，生产汽车的部分工序依次包括：焊接门框，安装车门，安装电动车窗。第一辆车进入焊接工序的同时，第二辆车进入装门工序（比第一辆车早一道工序），第三辆车进入装窗工序（比第二辆车早一道工序）。接着，所有汽车通过运输带自动进入下一道工序：第一辆车进入装门工序，第二辆车进入装窗工序，同时，另有一辆车接替第一辆车，进入焊接工序。计算机处理器的流水线技术也是一样的道理：将一条或一组指令的执行过程拆分为多个步骤，然后通过硬件处理单元尽可能多地并行执行这些步骤。处理器的流水线段数越多，在理论上可以并行执行的指令数也就越多。

除了流水线作业以外，还有其他方法可以让计算机并行执行指令，比如采用向量处理器。向量处理器不仅可以执行向量计算（一次直接操作一维数组，而不仅仅是一个数据），还可以并行运行多个处理器。近年来，在这些设计理念的指导下，许多“超级计算机”相继问世。2010年底，

中国研制出了世界上运算速度最快的计算机——天河-1A超级计算机。它配备了14,336颗至强X5670处理器（六核）和7168块NVIDIA Tesla M2050高性能计算卡（拥有448个流处理器核心），相当于一台计算机上安装了3297,280个处理器。在当今时代，主流计算机的特点是小巧、便宜，更新速度快，形同一次性产品。但即便如此，仍有一部分计算机与之相反。天河-1A共耗资8,800万美元（6亿元人民币，此处采用国内数据资料），占地面积约1000平方米，每年需耗资2,000万美元（约合1.4亿元人民币），以维持运行。天河-1A的浮点运算^[26]速度高达每秒2,566万亿次，是首批运算速度达到千亿次级别的计算机之一（日后势将有更多的计算机达到这个级别）。

这些体型庞大的计算机是人类设计史上的奇迹。不过事实上，在当今时代，就连个人计算机的处理器中，也包含人类有史以来最精巧繁复的设计作品。它们已经复杂到无以复加，以至于没有其他计算机的辅助，就不可能完成设计过程。无论是安排硅芯片上晶体管的布局架构，还是设计处理器的集成电路，这类低端的设计工作现在已经基本上没有人在做了。未来计算机的细节设计已经交给当今时代的计算机来承担。

并行化是计算机的未来

计算机技术的进步看似永无止境，而且，摩尔定律可能会让人想当然地以为，处理器势必会一直朝着更小巧、更便宜、更快捷的方向发展下去。但是，任何事物都不可能永远保持飞速发展的状态。事实上，我们在前进的道路上已经遇到了一个障碍。

计算机体系结构教授戴维·帕特森道出了其中的缘由：“过去十年到十五年的时间里，我们为了提升计算机的性能，不断地增加晶体管的数量。每一次增加晶体管的数量，都会使硅芯片的功耗和散热压力更大。每块芯片约100瓦的功耗已经是其散热能力的极限。我们大概在2003年达到了这个水平。要想继续利用摩尔定律提升计算机的性能，唯一的出路就是制造并行计算机

[\[27\]](#)。这就意味着我们必须改变编程模型

[\[28\]](#)，这是六十年的计算发展史上最重大的变革。”

也就是说，问题并不在于摩尔定律。因为在未来的许多年里，硅芯片上的晶体管数目似乎还会越来越多。真正的问题在于，一个体积较大、而且精密复杂的处理器会在运转的过程中发热。笔记本发烫到致人三度烧伤、处理器发烫到融化电路板，这种夸张的事情没有人愿意看到。唯一的解决办法是，制造更小、更简单的处理器，使

单个处理器的功耗减小，同时在单个芯片上集成多个处理器。计算机处理器的设计理念已然开始发生变迁。因此，现在市面上的计算机的时钟频率（clock speed）（用于衡量计算机解读新指令的速度）可能并没有快多少，但是处理器中的核数却增加了不少。

不过这里面有一个问题：既然有了并行或多核处理器，就理应配备能够高效利用其性能的软件，但是编写这样的软件对于程序员来说，是一件极其困难的事情。如今，硬件条件已经齐备，但是软件设施明显跟不上前进的步伐，无法充分发挥计算机的全部运算能力。帕特森同意这一观点，他在最近发表的一篇文章中写道：“处理器的并行化和微型化是计算发展史上的一个里程碑。”

“但这也是棋行险招之举，”他表示，“我们走过了探索之路的一个转折点，但却不知道前途是否光明。不过，不管怎么说，我们已经选择了一条前进的道路。我相信，计算领域的变革近在咫尺。”

“虽然并行化并不是万全之策，毕竟我们还没有取得实质性的突破，以简化并行编程的方法，”帕特森写道，“但是我们已经陷入绝境，实在没有别的出路可走，只能孤注一掷，希望将来有人能够写出好的并行程序，将硬件的性能充分

利用起来。”

“我想用万福玛利亚传球（Hail Mary path）来打个比方，”彼得森接着表示，“万福玛利亚传球是美式橄榄球术语，指成功率很低的长距离直传，一般在比赛快结束时候使用，孤注一掷地把球传出去，希望有人能在最后的时刻接球得分，剩下的就只有祈求圣母玛利亚保佑了。我们现在的情况就是这样。把球扔出去很容易；闭上眼睛用力扔球，然后祈求圣母玛利亚保佑，这也很容易办到。真正难的地方是接球。现在我们已经选择了并行化的发展道路，相当于把球扔到了空中。接下来就看有没有人能够引领技术创新，把球接住了。整个行业的未来就靠这样的人了。目前是一个非常关键的时期。”

有许多计算机科学家试图接住帕特森的万福玛利亚传球，但是到目前为止，问题尚未解决，我们还不知道怎样编写并行软件，才能使之与新的计算机体系结构相适应。怎样才能写出并行文字处理器或并行电子邮件程序？这个问题着实令人头疼。不过，值得庆幸的是，有一些应用程序在本质上讲就已经是并行程序了。其中最明显的例子，或许莫过于我们在每一台现代计算机上都会看到的、神奇的计算机图形。计算机图形往往由成千上万个微小的多边形拼接而成，多边形的表面覆盖着照片质量的图像。打个比方，假如你

想制作某件物品的三维模型，可以先用铁丝网将其形状构建出来，然后在表面覆盖图像，用以表现物品表面的图案。计算机图形也是一样的道理。无论是展示动画效果（比如游戏角色的动作、行驶在车道上的汽车），还是纯粹显示窗口和图标，计算机都在同时改变成千上万个多边形和图像的位置，而且每个多边形和图像的位置改变方式都非常相似。由此可见，计算机不费吹灰之力，就可以并行处理这些计算过程，提升运转速度。事实上，时至今日，图形的并行处理已经变得轻而易举，以至于大多数最先进的多核计算机结构都是图形处理单元（graphics processing units，简称GPU）。这些处理器已经有数百个内核，所有的内核都并行计算。因此，大多数游戏机、个人计算机、乃至小型便携式计算机都采用了并行GPU，以使图形流畅逼真。正因为GPU运算如此之快，内核如此之多，它已成为许多超级计算机的重要部件。

处理器的并行化趋势并不仅仅表现在单个芯片上。云计算是最近出现的一个新概念。它提供了一个动态虚拟的架构，这个架构或许会改变我们对计算机的认识。有了它，计算机用户就可以购买处理时间，使用异地多台计算机的软件和存储器，而不需要知道提供服务的计算机位于何处、其部件究竟如何运转。从概念上讲，云计算

将计算机视为一种资源——这跟水电的性质是一样的，我们在日常生活中都会用水用电，但却不需要知道自来水厂和发电站在哪里。云计算可以让用户使用最新软件，执行高强度计算，享受虚拟主机⁽²⁹⁾服务，而不需要参与异地物理主机的升级和维护。在线购书网站亚马逊就是云计算领域的创新先驱。该公司意识到，其庞大的数据中心通常只有10%的容量得到了有效利用，因此，在2006年，该公司开始推出亚马逊网络服务

（Amazon Web Services），出售数据中心的闲置容量。其他公司可以按需购买亚马逊提供的任何计算服务、软件和存储空间，而不需要维护或升级任何计算机。这一做法已变得越来越受欢迎，很多公司将来或许会从“云端”购买企业所需的一切计算服务，而不是大动干戈地建立和维护自己的内部计算机系统。另一方面，用户或许也可以通过为数众多的云端计算机执行并行处理，而不需要担心任务怎样拆分——只要云端软件足够智能，可以代劳就好。

处理器的并行化趋势还体现在汽车上，这里的并行处理比较容易实现。一台现代汽车上可能会搭载一百多个微处理器，它们协同工作，共同确保发动机和传动装置的平稳运转，同时控制仪表盘、车门锁、倒车雷达、车载收音机、GPS、车灯、后视镜、座椅调节器……事实上，大多数

汽车都有自己的计算机网络，使车内不同的计算机能够高效配合。在这一方面，读者朋友们或许已经有了亲身体验。当你踩下油门，发动机猛然加速时，车载收音机的音量也会自动变大。有些汽车甚至更加智能，可以将安全气囊加速计、停车灯、GPS导航系统、手机和车门锁进行互联，万一出了什么严重的事故，车子就会呼叫应急号码，将你的GPS坐标发送出去，同时解锁车门，打开停车灯。

超越冯·诺依曼

不过，在有些人看来，如此惊人的技术进步依然远远不够。曼切斯特大学教授史蒂夫·弗伯（Steve Furber）在职业生涯的开始，曾为艾康[\[30\]](#)（Acorn）计算机公司设计ARM 32位微处理器。这个设计在世界各地受到了热捧，如今，ARM内核的出货量已超过200亿个，比英特尔芯片的销量还多（ARM处理器本身也更昂贵）。多年来，全世界90%以上的手机都至少搭载了一个ARM内核。不过，弗伯虽然是正统的计算机设计师出身，但在1998年却决定改变方向。他意识到，生物大脑似乎比计算机处理器的计算和存储部件要优越许多。“最后，我一咬牙，就转方向了，”弗伯表示，“管他呢，反正我感兴趣的是神经网络[\[31\]](#)。”

如今，有越来越多的计算机设计师开始研究仿生计算机，弗伯就是其中的一员。他有一项雄心勃勃的设计，称为SpiNNaker。SpiNNaker是一个通用脉冲神经网络⁽³²⁾结构，其中包含成百上千个并行运转ARM微处理器。弗伯并不是唯一一个具备前沿思想的计算机科学家。如今，各种类型的仿生处理器正在紧锣密鼓的研制当中，它们的性能有的类似于大脑神经元，有的类似于免疫细胞，有的类似于细胞中的基因。有些研究人员甚至在尝试用新的材料代替硅，来制作处理信息的芯片。现在，还有诸多问题等着科学家来解答：我们能否在DNA链中存储信息，让它们跟基因一起重组？我们能否对细菌进行基因改造，从而发明新的计算机？我们能否发明量子计算机，让它根据匪夷所思的物理学量子效应来执行计算？

约克大学（University of York）教授安迪·泰瑞尔是一名计算机工程师，专攻仿生计算机。他认为，在未来的一段时间里，计算机体系结构依然会采用经典的冯·诺依曼设计。“但是，你可以想象，有‘一小撮’计算机工程师已经在设计新的体系结构了（希望他们的设计不仅新奇，而且令人振奋）。研制过程中或许采取了新的材料（或材料组合），比如分子器件、忆阻器⁽³³⁾（memresister）、生物电材料、各种化学结构

（液晶已经被别人试过了）。或许还有些新的材料我们根本就不知道。”

泰瑞尔认为，自然界中一定隐藏着某些秘诀，能够启发我们改善计算机的性能。毕竟，自然界中存在许多复杂而又高度精巧的事物。“怎样才能保持原有优势的基础上，将材料换成新的呢？这是一个难关。采用生物材料的系统似乎能搭载更多（也更复杂）的部件，其数量（和复杂度）可能会超乎人的想象。因此，我们面临的一大挑战或许是：怎样才能制造这样的系统？应该采用什么材料？怎样才能发挥它的性能？”

诚然，我们也许还未扫清前进道路上的所有障碍，但是，几乎没什么研究领域能像计算机科学这样发展迅速。到目前为止，还没有任何技术能与计算机在发展速度和效果上相提并论。

可以肯定的是，未来的计算机体系结构将向着速度更快、价格更低、并行性更强的方向迈进。

从早期的行业先驱到如今的电子工程师和计算机设计师，一代又一代天赋异禀的顶尖人才发挥天马行空的想象，投身永无止境的研究，才成就了如此非比寻常的成果——计算机。它们无处不在，面面俱到地服务于我们的生活。设计计算机，就相当于改变这个世界。

注释

(1) 摩尔定律是由英特尔（Intel）创始人之一戈登·摩尔（Gordon Moore）提出来的。其内容为：当价格不变时，集成电路上可容纳的晶体管数目，约每隔18个月便会增加一倍，性能也将提升一倍。换句话说，每一美元所能买到的电脑性能，将每隔18个月翻两倍以上。这一定律揭示了信息技术进步的速度。

(2) 1925年，AT&T总裁华特·基佛德（Walter Gifford）收购了西方电子（Western Electric）公司的研究部门，成立了一个叫做“贝尔电话实验室公司”的独立实体。AT&T和西方电子各拥有该公司的50%。贝尔实验室的工作可以大致分为三个类别：基础研究，系统工程和应用开发。

(3) 电话交换机是一种特殊用途的用户交换机。它有若干电话机共用外线，适用于机关、团体、中小企业等单位，也可以用于住宅和秘书电话。集团电话交换机不需要专职的话务员和维护人员，每部都可以通过指示灯了解整个系统的工作情况。当外线呼入时，可由任意一部话机应答，并可以转给所需的被叫。任何一部话机要呼外线时，只要按下代表空闲外线的相应键，即可拨号呼叫外线。

(4) 在逻辑中，真值或逻辑值是指示一个陈述在什么程度上是真的。在计算机编程上多称作布尔值。在经典逻辑中，唯一可能的真值是真和假。但在其他逻辑中其他真值也是可能的。

(5) 真值表是用于逻辑中（特别是在连结逻辑代数、布尔函数和命题逻辑上）的一类数学用表，用来计算逻辑表示式在每种论证（即每种逻辑变量取值

的组合)上的值。

(6) 所谓开关电路是指具有“接通”和“断开”两种状态的电路。

(7) 普林斯顿高等研究院，1930年成立于美国新泽西州普林斯顿。它并不是普林斯顿大学的一部分。在这里，各个领域的科学家可以开展最纯粹的尖端研究，而不受任何教学任务、科研资金或者赞助商的压力。在其他地方也有基于普林斯顿高等研究院而建立的高等研究院。

(8) 赫尔曼·克劳斯·胡戈·外尔（Hermann Klaus Hugo Weyl, 1885年11月9日—1955年12月8日）是一位德国数学家。他传承了以大卫·希尔伯特和赫尔曼·闵可夫斯基为代表的哥廷根大学学派的数学传统，为理论物理上和在纯数学领域（如数论）做出了杰出的贡献，是20世纪最有影响力的数学家之一。

(9) 第一次世界大战期间，英国组织了由一群战舰保护的跨洋护航。第二次世界大战期间，相同的系统保护了同盟国的船只免受德国潜水艇的袭击。

(10) 康拉德·楚泽（1910年6月22日—1995年12月18日），是一位德国工程师和计算机先驱。

(11) 条件分支依据一个条件式的测试结果为真值或伪值，而将流程将转移至相关连的程式区块。在许多常见的编程语言中，条件分支是一种叙述，它们决定哪段程式区块被执行但本身没有值。

(12) 赫尔曼·戈德斯坦曾经在马里兰阿伯丁试验场的弹道实验室担任军械数学家，弹道实验室曾征用摩尔电气工程学院的计算设备。

(13) 神经元按照不同的功能，可分为联络神经元、感觉神经元和运动神经元。联络神经元是指中

中枢神经系统中将感官与运动神经细胞连在一起的神经细胞。感觉神经元又称传入神经元，会将神经刺激由受体或感觉器官传送到中枢神经系统。运动神经元又称传出神经元，会将中枢神经系统的兴奋传到各个器官或外围部分。

[\(14\)](#) 见注释40

[\(15\)](#) 见注释40

[\(16\)](#) 寄存器是中央处理器内的组成部分。寄存器是有限存贮容量的高速存贮部件，它们可用来暂存指令、数据和地址。

[\(17\)](#) 输入接口和输出接口是电子设备之间数据传输的重要通道。

[\(18\)](#) 洛斯阿拉莫斯国家实验室，隶属美国能源部的国家实验室，位于新墨西哥州的洛斯阿拉莫斯。1945年7月，该实验室研制首枚原子弹，曼哈顿计划的所在地，在冷战时扮演美国军事研究的重要基地。

[\(19\)](#) 是指用简单的硬件操作来模拟当时无法用技术直接实现的复杂指令。

[\(20\)](#) 德州仪器是世界上最大的模拟电路技术部件制造商，全球领先的半导体跨国公司。

[\(21\)](#) 计算机体系结构是程序员所看到的计算机的属性，即概念性结构与功能特性。

[\(22\)](#) 计算机协会是一个世界性的计算机从业员专业组织，创立于1947年，是世界上第一个科学性 & 教育性计算机学会。

[\(23\)](#) 磁盘阵列是由很多价格较便宜的磁盘，组合成一个容量巨大的磁盘组，利用个别磁盘提供数据所产生加成效果提升整个磁盘系统效能。利用这项技术，将数据切割成许多区段，分别存放在各个硬盘

上。

(24) 存储器可分为外部存储器和内部存储器。外存通常是磁性介质或光盘等，能长期保存信息。内存指主板上的存储部件，用来存放当前正在执行的数据和程序，但仅用于暂时存放程序和数据，关闭电源或断电，数据会丢失。

(25) 高速缓存是访问速度比一般随机存取存储器（RAM）快的一种RAM。

(26) 浮点运算简单地说就是小数的加减乘除运算。在计算机的硬件中是没有小数点这个东西的。cpu能处理的东西都是整的。所以，小数就要用类似科学计数法的方式来表示。如1.234在计算机里面，可以理解成用1234和-3两个整数来表示 1234×10^{-3} 次方，这类数就叫浮点数。

(27) 并行计算机是指有两个或两个以上的处理器连接起来可以并行操作的计算机。

(28) 编程模型模型又称为并行计算机，是指由程序员所看到的一个抽象并行计算机。

(29) 虚拟主机是在网络服务器上划分出一定的磁盘空间供用户放置站点、应用组件等，提供必要的站点功能、数据存放和传输功能。

(30) 艾康计算机公司是现在著名的ARM公司的前身。它创立于1978年，位于英格兰剑桥，80年代为个人计算机领域的革新做出了重要贡献，相当于英国的“苹果公司”。

(31) 人工神经网络（artificial neural network，缩写ANN），简称神经网络（neural network，缩写NN），是一种模仿生物神经网络的结构和功能的数学模型或计算模型。

(32) 脉冲神经网络 (Spiking Neuron Networks, 简称SNN) 经常被誉为第三代人工神经网络。第一代神经网络是感知器, 它是一个简单的神经元模型并且只能处理二进制数据。第二代神经网络包括比较广泛, 包括应用较多的BP神经网络。但是从本质来讲, 这些神经网络都是基于神经脉冲的频率进行编码。借助于神经科学的研究, 人们可以精确地建立基于脉冲产生时间神经网络模型。这种新型的神经网络采用脉冲编码 (spike coding), 通过获得脉冲发生的精确时间, 这种新型的神经网络可以进行获得更多的信息和更强的计算能力。

(33) 忆阻器又名记忆电阻, 是一种被动电子元件。如同电阻器, 忆阻器能产生并维持一股安全的电流通过某个装置。但是与电阻器不同的地方在于, 忆阻器可以在关掉电源后, 仍能“记忆”先前通过的电荷量。

二进制数字中的生活

虽然人的思想、财富、记忆乃至娱乐活动对于计算机来说都是镜花水月，但是计算机的思想并不简单，它和人的心理机制一样，分为不同的层面。最低级、最原始的层面是机器的本能，最高级的层面涉及到对抽象概念的识别和理解。介于两者之间的，是硅芯片处理器的算术运算和逻辑运算活动。

计算机和我们不一样，它们在每一个思想层面上都有相应的语言。要想给它们灌输新的理念，只要在任何一个层面上下功夫，都可以达到预期的效果。我们可以输入指令，让计算机识别和理解宏观抽象、而又错综复杂的概念，然而，一旦指令中出现任何细小的错误，都有可能让我们的数字奴隶在刹那间全盘崩溃，就像癫痫症发作一样。

既然唯命是从的计算机不懂得灵活变通，怎样构建其思维方式，才能使计算机成为我们值得信赖的可靠助手？一旦它们的思想复杂到超乎人的想象，怎样才能确保它们像原来一样服从命令？

阳光透过讲堂的大窗户倾泻进来。近三十位声名卓著的工程师和科学家正仔细聆听着约翰·莫奇利在台上所说的每一句话，他们不时地记着笔记，讲堂里响起一片写字的沙沙声。听讲的人当中，有一个人来自赖特-帕特森空军基地（Wright-Patterson Air Force Base，俗称Wright Field）的军械实验室（Armament Laboratory），他叫加尔（Gard）。加尔在众人当中显得尤其勤奋，笔记

已经洋洋洒洒地写了几百来页。

这是1946年的一个温暖和煦的仲夏日，时值星期一早晨。回想起三年前与图灵在下午茶时间的热烈讨论，克劳德·香农觉得恍如隔世。此时此刻，他正在宾夕法尼亚大学的摩尔电气工程学院学习一门为期八周的课程，这门课目前已经上到了第三周。能够受邀来此听课，对看他来说已是一大荣幸，毕竟，这是少数人才有的殊荣。课程采用讲座的形式，主题是电子数字计算机的设计。这是世界上第一门关于计算机科学的课程，香农发现，课上所学的许多思想在他的头脑里擦出了灵感的火花。他最近从莫奇利那里学到了一个新词——“编程”。它通常作为动词使用。给电子计算机编程是一个让人耳目一新的概念。香农还听到了一些关于办公室政治的八卦：给他们开课的两位讲师——莫奇利及其同事埃克特四个月前刚从宾夕法尼亚大学辞职，原因似乎是围绕“埃德瓦克”产生的专利纠纷。另一名讲师戈德斯坦近期就要去高等研究院（IAS）入职了，届时他将与香农的老同事约翰·冯·诺依曼共事。冯·诺依曼本来也要到这里来开设两个星期的讲座，但是他好像不会来了，因为临时有事。

克劳德·香农调整了一下坐姿，弄得椅子咯吱作响。他环视了一下整座讲堂。约翰·莫奇利在台上讲得正投入，声音清晰而自信。今天的课堂内

容是：二进制与十进制数字系统之间的转换。尽管香农对这方面的内容已经了如指掌，但他依然兴趣浓厚，因为他正在进行密码学研究。

在一篇机密级的论文中，香农发布了他最近在贝尔实验室开展密码学研究的部分成果。这些研究成果与他当年攻读理科硕士期间提出的思想一脉相承，只不过二者的着眼点不同。香农当年着眼的是开关电路，现在着眼的则是密码学背后的数学原理。他开始意识到，破译扰频[\(1\)](#)加密的信息，其实大体上就相当于给正常传输的信息纠错。举个例子，如果一段编码中出现了重复信息——比方说它的内容中包含许多常见的词语，如“这个”、“一个”、“和”，那么这段编码破译起来就会轻松许多。因为我们知道，“这个”、“一个”、“和”这类简短的词语是句中常见的语法成分，就算把它们的每个字都改头换面，也不难根据它们在句中出现的位置来判断其成分。只要破译了高频词，那么整条加密信息指的是什么内容，或许就能猜出个大概了。因此，要想增加破译难度，必须尽量减少冗余信息。但是，如果你想尽可能多地保留原始信息，而记忆存储系统又很容易出错，那就最好少删除一些冗余信息。

检验信息是否出错的方法之一，就是设置奇偶校验位（parity）。至于奇偶校验位是指什么，还得从比特的概念说起。比特（bit）是内存中的

最小单位，也称作“位”、它只有两个状态，分别以1和0表示。我们将8个连续的比特叫做一个字节（byte），比如（1、0、0、1、1、1、1、0）就是一个典型的字节。如果其中某一位存储了错误的值，那就会导致信息出错。为了检测信息是否出错，我们在每一个字节（8位）后面又额外增加了一位，称为奇偶校验位。这样一来，原来的8位字节就变成了新的9位字节。奇偶校验位也只有1和0两种值。如果原字节中1的个数为奇数，那么奇偶校验位就设为1，这样一来，新字节中1的个数就变为偶数；反之，如果原字节中1的个数为偶数，那么奇偶校验位就设为0，这样一来，新字节中1的个数依然为偶数。也就是说，凡是带有奇偶校验位的字节当中，1的个数始终应该为偶数，如果你发现某个字节不是这样，那就说明它有错误，这段字节包含的信息就需要重新读取。（这个方法在20世纪50年代早期开始在计算机领域广泛采用，后来，人们很快就开发出了更多高级的方法。）

言归正传，香农在摩尔学院听讲座的过程中，了解了二进制对于计算机的重要价值：二进制数不仅是计算机内各个部件交换信息的重要载体，还是确保信息存储和检索过程不出差错的重要工具。后来，香农发明了术语“比特”来指代二进制数位（binary digit），同时阐述了如何利用

比特来衡量信息量，并对信息进行传输、加密、压缩、纠错。

在当年赴摩尔学院听讲座的人当中，香农并不是唯一一个计算机领域的先驱。莫里斯·威尔克斯也去了摩尔学院，只不过他差点就错过了这门课程。香农在台下听莫奇利讲解二进制与十进制数的那一天，威尔克斯还在英国。

威尔克斯1913年出生于英国达德利市⁽²⁾（Dudley），从小体质不好，饱受哮喘和过敏反应的折磨，生活受到了很大的影响，以至于在八岁入学后，不幸在差等生中垫底。校长在学年末给他的评语中用红笔写道“常年缺勤旷课，成绩上不去”，威尔克斯成了少数需要留级一年的学生之一。不过留级后，威尔克斯发现自己的身体状况和学习成绩明显好了许多，而且自此之后，就再也没有跌回过去的低谷。他终于在入学四年来第一次融入了自己的学校，内心的激动之情难以言表。此外，出于对数学和科学的热爱，他对校园的学习生活更是满腔热忱。威尔克斯一直痴迷于这两门学科，他说：“我从来没有怀疑过自己将来会成为科学家或工程师。”他最喜欢玩机械玩具，小时候喜欢花大笔的时间捣鼓电池、灯泡和门铃，长大一点后，他自制了矿石收音机⁽³⁾，还从《无线世界》（Wireless World）杂志上自学了电子学的知识。他在预科班修读了物理

学和数学，毕业后进入剑桥大学攻读数学。他和艾伦·图灵在同一时期学习了同一门课程，但是两个人似乎从未注意到对方的存在。1934年，两人大学毕业后，图灵继续深造，在马克斯·纽曼的指导下修读高级数学课程；威尔克斯则进入卡文迪许实验室⁽⁴⁾（Cavendish Laboratory），研究极长波在电离层中的传播特性。与此同时，威尔克斯继续发扬他的动手实践精神，改良了一台由麦卡诺⁽⁵⁾（Meccano）金属部件拼装而成的微分分析仪模型，用于计算微分方程。不久后，他作为核心技术人员，为剑桥新建的数学实验室

（Mathematical Laboratory）采购和安装了一台大型机械式微分分析仪。1939年，他拿到了硕士学位，之后又拿到了博士学位。

由于战事一触即发，威尔克斯受邀参与了雷达的秘密研制工作。由于他在无线电应用数学领域经验丰富，因此自然被委以重任，在英国海岸线上安装新的雷达设备，以探测船只。战争时期，研制雷达成了他的职业。威尔克斯一开始供职于克赖斯特彻奇（Christchurch，新西兰第三大城市）的防空实验研究基地（Air Defence Experimental Establishment），研究试验雷达。1943年，他来到多塞特郡的沃斯麦特乐佛（Worth Matravers），在皇家空军的电信研究基地（Telecommunications Research Establishment）研

究雷达导航，以解决轰炸机的目标定位问题。战争结束后不久，威尔克斯自愿来到饱受战火摧残的德国，在多个科研中心走访了在押囚犯，学习他们在战时的研究成果。离开德国后，他回到了剑桥大学，投入自己喜欢的科研工作。

回到熟悉的环境后，威尔克斯意识到，自动计算机的研制工作刻不容缓。他听说了埃克特和莫奇利在美国研制埃尼阿克的工作，于是在1946年2月给学院提交了一份报告。他在报告中写道：“这是一个大有可为的研究领域，电子应用技术首当其冲。它在战时取得了迅猛的发展。美国人在这门学科上已经领先了一步，我觉得剑桥也应该迎头赶上。”

三个月后，机械计算机专家莱斯利·科姆里（Leslie Comri）来访剑桥。他从美国带来了一份手稿副本——即约翰·冯·诺依曼所写的《关于埃德瓦克的报告初稿》。威尔克斯只有一个晚上的时间阅读报告，当时还没有影印机，他只能边读边作笔记。威尔克斯很快就被报告的内容吸引住了。“我很快就意识到这个研究成果非同小可，”他表示，“从那以后，我对计算机的发展前景一直没有怀疑过。”就在威尔克斯依然对报告的内容记忆犹新之时，他突然接到了摩尔学院院长哈罗德·彭德（Harold Pender）发来的电报，邀请他去参加一门新开的电子计算机课程。开课时

间就在几周之后。当时，旅行依然受到严格的限制，威尔克斯担心他的赴美之行不会得到批准。开课时间是在7月8日，但他直到7月23日才拿到批准，可以乘坐一艘八月初出发的小船前往美国。尽管这个方案太不如人意，但威尔克斯别无选择，只能接受。

威尔克斯的越洋航程并不顺利。尽管伙食条件一流，但是住宿条件太差，35个人挤在一艘只能容纳20个人的小货船里。更糟糕的是，发动机在中途抛锚了好几次。就这样，威尔克斯一路历经磨难，终于在8月15日抵达了纽约，上岸后又马不停蹄地赶路，总算在8月18日赶到了费城。这时候，他已经错过了三分之二的课程，好在前面的课程大多都是些入门性质的讲座。到了8月19日，也就是星期一，威尔克斯抵达摩尔学院，正好赶上当天下半节课。这堂课讲的是埃尼阿克的细节内容，讲师提供了详尽的电路图。

访美期间，威尔克斯充分利用了每一个可能的学习机会，吸收计算机科学领域的前沿知识。他在课下与莫奇利探讨了埃尼阿克的结构细节，后来又专程拜访了戈德斯坦，弥补落下的课程。两人在共进晚餐期间进行了深入讨论。由于当时冯·诺依曼正在高等研究院研制新的计算机，两人探讨了这台新计算机的逻辑设计思想。此外，威尔克斯还前往波士顿，拜访了其他计算机设计师

——哈佛大学的霍华德·艾肯（Howard Aiken）和麻省理工学院的塞缪尔·考尔德韦尔（Samuel Caldwell）。艾肯发明了自动顺序控制计算器

（Automatic Sequence Controlled Calculator），在他的主持下，由继电器构成的“马克二号”[\(6\)](#)

（Mark II）自动计算机正在研制当中。考尔德维尔则用电机变速箱设计了一台改良版的机械式微分分析仪。

威尔克斯回到剑桥大学，满脑子都是计算机领域的前沿思想和美国人取得的一些关键成果。自从摩尔学院的计算机课告一段落后，听讲的工程师和科学家各奔东西，有的组建了新的公司，有的组建了研究团队，威尔克斯则担任了剑桥大学的数学实验室主任。当时的数学实验室[\(7\)](#)已经相当于一个完整的院系，科研经费十分充足。

威尔克斯认为，是时候研制一台实用的存储程序计算机[\(8\)](#)了。幸运的是，没过多久，餐饮巨头J. Lyons & Company就给他提供了科研经费和技术人员，因为该公司需要计算机进行会计核算，管理员工工资单。威尔克斯的项目进展很快，他在雷达领域积累的经验更是大大加速了这一进程。项目团队在此基础上构建了一个工作记忆[\(9\)](#)系统，用于存储数字。

威尔克斯在研制新机器的过程中，旁听了图灵开设的几个讲座，讲座的主题是图灵关于自动

计算机的设计思想。不过，两个人的设计理念并不一致。图灵认为，计算机应该在水银延迟线存储器⁽¹⁰⁾的基础上进行优化设计。威尔克斯的观点正好相反。“我认为，水银延迟线存储器迟早要被真正的随机存取存储器淘汰。与其把时间和精力都花在一项短命的技术上，还不如多下点功夫研究编程，毕竟，编程领域还有那么多问题值得研究。”图灵也不赞赏威尔克斯的设计理念，他在一篇备忘录中写道，威尔克斯有一些理念“比较偏向美国传统，遇到什么困难就喜欢依赖设备，而不喜欢动脑子。”

但是事实证明，威尔克斯的方法更加实际。剑桥大学的电子延迟存储自动计算机（Electronic Delay Storage Automatic Calculator，简称EDSAC）于1949年5月6日投入运行，直到1958年才光荣退役。它是世界上第一台实用的存储程序计算机。

学习计算机编程

剑桥大学的EDSAC计算机并不只是一台前沿尖端的机器，它还开创了计算机领域的先河。此前的计算机每次执行新的运算，都需要插入不同的线路进行重新装配，而EDSAC则通过存储器中的软件实现各种不同的运算操作，这就对编程提出了很高的要求。为了写出功能强大的软件，威

尔克斯和戴维·惠勒（David Wheeler）等研究人员提出并改进了许多新的思想，时至今日，这些思想在计算机编程领域已占据主流地位。

不过编程在当时并不是一件容易的事情。所有的早期计算机先驱很快就意识到，一旦设计出存储程序计算机，就必须拼了老命地编写计算机能够运行的程序。如果任何程序都无外乎是一组能够触发数学或逻辑学电路的二进制数（就像我们在上一章所看到的那样），那么编写软件就会变成一场噩梦——而事实也的确如此。1949年6月，惠勒第一次意识到了编程的艰难。他后来回忆起了当时的情形：“那时候，我正试着让自己编写的第一个真正意义上的程序运转起来。有一次，我像往常一样从EDSAC机房出来，准备去操作打孔机，突然站在楼梯转角处犹豫了，心里意识到，单是给自己的程序除错，可能就要花掉我大半辈子的时间。”

显然，当时的科学家需要一些新的思想，来简化编程过程，提高编程能力。对此，威尔克斯（在几年后）提出了一个方法，称为微程序设计。当时，麻省理工学院正在研制的旋风计算机 [\(11\)](#)（Whirlwind）给了他部分灵感，让他意识到，并不是每一条低级指令——比如除法——都需要电子电路来执行。复杂的指令完全可以分解成一系列简单的指令，而微代码编写出来的微程

序可以作为二进制机器代码和硬件之间的桥梁。事实证明，这一方法实用性很强，时至今日，复杂指令集计算机⁽¹²⁾（Complex Instruction Set Computer，简称CISC）处理器依然采用了这一思想原理，以执行高度复杂的操作。而精简指令集计算机（Reduced instruction set computer，简称RISC）处理器则不使用微编程。

但是即便是使用微代码，计算机编程人员依然需要编写一长串的数字，即机器代码指令。而且，早期的编程人员还因为机器的内存容量极其有限而备受掣肘。EDSAC的内存只有两千字节左右（放到今天，一部手机的内存都比它大几百万倍）。为了解决这些问题，威尔克斯的团队又想出了一个妙招——编写子程序。研究人员意识到，许多程序在运行的过程中，都需要重复执行某个操作——比如在某个复杂的数字运算中，需要多次进行开平方操作。如果每次开平方都得把平方根代码写上，那么程序当中就会出现许多重复代码，占用不必要的空间，使程序变得庞大而低效。这就好比你在写一个句子时，不仅构造了完整的语法结构，还将句中每个词语的定义也写了下来。比方说有个好端端的句子：

一只小猫坐在地毯上。

结果被你写成了：

一种以“只”为度量单位、称为“猫”的小型家

养猫科食肉动物，以臀部着物的方式坐在一种名为“地毯”的装饰地面的厚实织物上。

为了简化编程过程，威尔克斯的方法是建立子程序库，也就是将常见的函数[\(13\)](#)单独列出，集中起来，就像把常见的词语及其释义收录在词典中一样。一旦程序在运行的过程中需要使用到某个常见函数，计算机就会在子程序库中“查找定义”，执行相应的子程序代码，根据输入值进行运算，再将运算结果返回。

在这一方面，威尔克斯的理念已经领先于同时代的大多数人。冯·诺依曼有一次突然造访，与威尔克斯进行了讨论。两个人的观点产生了分歧。威尔克斯回忆道：“他觉得应该把开平方运算嵌入到计算机的指令集中……我自己的立场则有所不同，我已经把子程序看做是对基本指令集的扩展，所以觉得没有必要再在指令集中嵌入一个特殊的函数。”

由于威尔克斯领导的剑桥大学研究团队很早就开始在编程领域开疆拓土，他们对编程的艺术也颇有心得。普林斯顿高等研究院的冯·诺依曼喜欢绘制“程序框图[\(14\)](#)”，来展示程序应该如何运作。所谓程序框图，大体上就是一系列带箭头的步骤，可以展示控制流[\(15\)](#)。不过，威尔克斯手下最好的程序开发人员戴维·惠勒认为，要想写出出色的软件，还需要其他的方法。“我们关心的

首要问题是，用户在操作过程中是否得心应手，因此在这方面也下了很大的功夫。”他表示，“模块化设计⁽¹⁶⁾的编程风格很早就开始得到推行。这是我们开展编程教学的方式。当然，我们也知道一些其他的方法，比如冯·诺依曼的程序框图。不过这些方法一般都不用，就算用也只是拿来说明已经完成的步骤。我们发现，只要将复杂的问题进行分解，用一个个子程序加以解决，然后将子程序置于主程序的控制之下，就可以逐步形成模块化设计的思维方式，到时候程序设计自然水到渠成，根本不需要使用到程序框图。个人认为，程序框图最容易让人写出垃圾软件。并不是说这种方法毫无价值，只不过你要是想拿它来代替思考过程，根本就不管用。”

时至今日，利用子程序编程的思想已被普遍接受，几乎所有的编程语言都用到了这个概念。现代的程序员在编程过程中都会用到函数库，正如1949年研制EDSAC的科学家一样。任何其他的编程方式在今天都是无法想象的。

威尔克斯的团队还考虑到了程序员可能遇到的其他困难。他们从一开始就意识到，尽管计算机只能理解数字，但很少有人能够在只采用数字的情况下编写程序，了解计算机的运行过程。人的大脑习惯了阅读文字和符号，而不是处理一连串数字。惠勒认为，冯·诺依曼在高等研究院研制

的计算机在这一方面做得很差，完全比不上剑桥大学的EDSAC。“它太原始了，这让我非常震惊，”他表示，“我估计它的程序是用二进制输入的。我们剑桥大学的研究团队很早就开始采用了一种叫做‘汇编程序[\(17\)](#)’的工具。”它能够转换编程语言，从十进制转为二进制，使用助记符[\(18\)](#)，可引用代码，可分隔字段，可自动定位子程序，还具备其他各种功能。我们基本上从开展项目的第一天起就已经在使用汇编程序了。这极大地简化了编程过程。

这样一来，程序开发人员就不需要和抽象的二进制数打交道了，他们可以采用简短的词语来编写程序，这些词语看起来有点像英文单词，以下的代码为例：

```
    cmpl $7, -4( %ebp)
```

```
    jle  .L6
```

```
    jmp  .L4
```

这段代码的意思是：将变量值与数字7的大小进行比较；如果小于或等于7，则跳转到程序中标为“L6”的区域；如果大于7，则跳转到“L4”区域。

与一连串抽象的二进制数字相比，这段奇怪而又晦涩的文字无论在阅读还是理解上，都要轻

松许多。它与处理器使用的低级的代码没有太大的分别：其中的每一个词语（或命令）——比如“cmpl”、“jmp”——都直接对应于机器代码中的一条指令。要将汇编语言写成的程序转换为相应的机器代码，就需要使用到另一种计算机程序，称为汇编程序。汇编程序读取的是用汇编语言书写的源程序，输出的是用机器语言表示的目标程序。

汇编语言在计算机领域举足轻重，20世纪90年代的所有程序都是由汇编语言写成。几十年来，有两种常见的程序只采用汇编语言：一是计算机游戏（因为开发人员希望尽可能地提高游戏的运转速度，同时尽可能给玩家带来极致丰富的游戏体验），二是操作系统。即便是在当今时代，程序员要想编写速度超快而形式简洁的代码，都免不了要采用一些汇编语言。

攀登更高峰

1951年，计算机开始搭载好几个层次的软件。第一层是微代码，它完全依赖于芯片内部的硬件连接。第二层是机器代码，它比微代码更抽象一些。第三层是汇编语言，它比机器代码可读性稍强。

计算机编程，说白了其实就是告诉计算机应该使用哪种逻辑和算术电路。要想给计算机编

程，程序员可以使用汇编语言来写代码，而汇编语言正如我们在上一节所看到的那样，和英文单词有些类似。这些代码随即由汇编程序转化为机器代码，机器代码定义微程序的指令，微程序则在算术和逻辑单元（ALU）的电子元件中被翻译为一系列指令的组合。

不过，汇编语言对于许多程序员来说依然难度太大。如果你想处理更复杂的思想 and 概念，那么，纠结于个别的跳转指令只会拖累你的步伐。如果你希望自己的程序能够在完全不同的处理器上运行，那就需要采用高级的编程语言，也就是独立于底层硬件的计算机语言。

1953年，随着电子计算机在全世界遍地开花，发明抽象编程语言的问题开始受到广泛关注。莫里斯·威尔克斯受邀主持了美国计算机协会（ACM）在麻省理工学院召开的一期研讨会，专门探讨这个问题（当时的会议主题为自动编程）。威尔克斯至今还清楚地记得当年会上的讨论情况。“与会者的意见分歧相当尖锐。有些人认为，凡是试图绕开困难的做法都是误入歧途。程序员只有老老实实地恪守本分，编程领域才会取得更大的发展。另一方面，还有一些人认为，只有新的技术才具有实实在在的实用价值。”（时至今日，计算机科学家当中依然存在这两派的纷争。）

许多研究人员已经在试验新的、更简单的编程语言。早在1949年，约翰·莫奇利就发明了一种语言，称为简代码（Brief Code），后来更名为短代码（Short Code）。短代码虽然使用起来简单许多，但是需要翻译——也就是说，计算机每次运行这种语言编写的程序，都得临时将短代码翻译成机器代码。这就意味着，这种程序的运行速度比机器代码写成的程序慢54倍。与此同时，在英国曼彻斯特，一位名叫埃里克·格伦尼（Alick Glennie）的研究人员发明了另一种语言。

这种语言易于使用，可通过另一种程序自动转换为机器代码，因此具备简单实用、运行速度快的双重优势。格伦尼将其称为自动代码（Autocode）。以下是一个自动代码程序中截取的部分语言片段：

```
n1 = 0
```

```
13) PRINT v(300 + n1), 4024
```

```
n1 = n1 + 1
```

```
-> 13, n1 # n20
```

```
TEXT
```

```
NUMBER OF LIQUID MOLES =
```

```
PRINT v215, 4040
```

```
STOP
```

显然，用自动代码写程序比用汇编语言要轻松许多。其中有些自动代码语句和好几条机器代码指令相对应，不过程序员不需要为此挂心。只要使用另一种称为编译器的程序，就可以将这段由自动代码编写而成的简洁英文命令翻译成机器代码。自动代码是世界上最早出现的编译型高级编程语言。

自计算机协会在麻省理工学院召开研讨会，讨论自动编程问题后，越来越多的计算机科学家开始意识到，编程过程的简化势在必行。1957年，IBM的约翰·巴库斯（John Backus）发明了另

一种编译型高级编程语言，称为福传（FORTRAN）。福传的全名是Formula Translation，意思是“公式翻译”。这种语言甚至比自动代码更高级，可以用来编写更加复杂的程序。它的编译器也极为智能，可以生成非常简洁高效的机器代码。以下是福传程序的一个例子——它是一个排序子程序：

```

1      SUBROUTINE SORT (A, N)

      DIMENSION A (150)

      DO 4 J = 2, N
      SMALL = A (J - 1)

      DO 4 I = J, N

      IF (A (I) - SMALL) 3, 4, 4

3      SMALL = A (I)

      A (I) = A (J - 1)

      A (J - 1) = SMALL

4      CONTINUE

      RETURN

5      END (2, 2, 2, 2, 2, )

10     DIMENSION ALPHA (150), BETA(100)

```

没过多久，许多适用于其他计算机的福传编译器相继问世，这样一来，同一款福传程序就可

以编译成不同计算机所特有的机器代码。从这一刻开始，程序员便有了一种新的工具——“便携式”代码，它可以使同一款程序在完全不同的计算机上运行。很快，其他编程语言也开始纷纷效法，比如算法语言（ALGOrithmic language，简称ALGOL）、列表处理语言（LISt Processor，简称LISP）、初学者通用符号指令码（Beginner's All-purpose Symbolic Instruction Code，简称BASIC）。随着计算机设计师开始研制晶体管计算机，这些语言也得到了稳步改进。

很快，计算机科学家便能分析编程语言，并用数学方法——包括邱奇的 λ 演算（详见第二章）将其形式化^{[\[19\]](#)}。程序员也可以在代码中表达更抽象的概念，而不需要操心低层次的细节问题。各种类型的编程语言相继问世。早期语言基本上都是过程式语言（程序员告诉计算机如何执行过程步骤），后来的语言则采用了不同的编程范型。在面向对象的语言^{[\[20\]](#)}（Object-Oriented Language）中，数据及其操控方法都封装^{[\[21\]](#)}在“对象”中，以实现代码的模块化，防止数据的意外损坏（这一点也是程序的“副作用”）。函数式编程语言利用若干简单的执行单元让计算结果不断渐进，逐层推导复杂的运算，而不是像过程语言一样，设计一个复杂的执行过程。此外，还有更多适用于并行计算机的程序语言相继问世。

正因为众多早期先驱的开创性工作，如今人们习以为常的一些重要编程思想才得以诞生。威尔克斯继续投入计算机语言的研究，他在改良算法语言60（ALGOL 60）的基础上发明了CPL（Combined Programming Language）编程语言。这种语言并没有受到热烈的反响，但是却奠定了BCPL（Basic Combined Programming Language）语言的基础。BCPL进一步发展演变，推动了B语言和C语言的问世。直到现在，C语言（以及在此基础上形成的诸多语言，如C++、C#、Objective C）或许是世界上应用最广泛的计算机编程语言之一。人们当前使用的许多众所周知的操作系统（比如UNIX、Linux、Mac OS X、Windows）都是用C语言写成的。如今，几乎每一台计算机上都搭载了C语言编译器，方便用户使用C语言编写代码。以下列出的快速排序函数（就是我们第二章看到的那种快速排序）正是用C语言写成的：

```

void quicksort(int arr[], int low, int high)

{ int i = low;

  int j = high;

  /* compare value */

  int z = arr[(low + high)/2];

  /* partition */

  do

  {   while(arr[i] < z) i ++; /* find member above */

      while(arr[j] > z) j--; /* find element below */

      if (i <= j)

      {swap(arr[i],arr[j]); /* swap two elements */

        i ++; j--;

      }

  } while(i <= j);

  /* recurse */

  if (low < j) quicksort(arr, low, j);

  if (i < high) quicksort(arr, i, high);

}

```

莫里斯·威尔克斯（如图5）继续为计算机科学领域做出贡献。1956年，他当选为皇家学会的

成员；之后创立了英国计算机协会（British Computer Society，简称BCS），并在1957至1960年期间担任第一届会长。1967年，他因为在EDSAC计算机上的研究工作而获颁图灵奖（Turing Award）。2000年，他被女王授予了爵士勋衔。2010年11月，威尔克斯在家中平静逝世，享年97岁。



图5 莫里斯·威尔克斯爵士与世界上最早的、仍在运行的电子计算机——1951年在哈维尔原子能研究基地（Atomic Energy Research Establishment）投入使

用的“哈维尔·德卡特伦”（Harwell Dekatron），2009年摄于布莱切利公园，国家计算博物馆。

“第一个研制出存储程序计算机的科学家是莫里斯·威尔克斯。他带领的团队提出了许多重要思想，发明了许多软件和硬件。我得说，威尔克斯给计算机科学带来了重大影响。”戴维·帕特森表示。

计算机历史学家西蒙·拉文顿（Simon Lavington）教授也对威尔克斯的重要性表示认同。“剑桥大学教授的莫里斯·威尔克斯爵士是英国当之无愧的计算之父。”

数据库

随着编程语言的功能日渐强大，计算机的速度日益加快，容量日益扩大，可靠性日益增强。很快，计算机科学家就意识到，除了计算方程式的结果，计算机还可以在更广阔的领域大有可为。它们可以快速存储和处理大量的数据。

如果只是单纯存储一系列数字，那倒不是什么难事，但是，如果存储的是病历记录或者财会记录，那么情况就会复杂许多。这就要求计算机必须能在短时间内迅速找到特定的信息条目——比如某一份病例中记述的病因，或者某一天产生的支付记录。怎样编排数据才能帮助我们轻松地

找到所需信息呢？

这些都不是什么新的问题——自从第一部词典问世以来，人们就面临着快速查找词条的问题。其解决办法，就是按照字母表的顺序编排词语。以某个单词为参照，词典中的其他词语中要么在它之前，要么在它之后。因此，我们在查找单词时，可以翻开词典的任何一页，然后根据上面的词语决定应该往前翻还是往后翻，直到将我们想查的单词查找出来。

第一代数据库都是按照非常类似的原则编排的，它们于20世纪60年代问世，被称为导航式数据库管理系统（*navigational database management system*）。数据库第一次打开时，就会自动出现一个指针[\(22\)](#)，指向第一条存储记录（好比翻开词典正文的第一页，就会看到第一个词条是以a开头的单词）。这些存储记录按照链表[\(23\)](#)或网状结构[\(24\)](#)编排，每一条记录当中都存有一个指针，指向相邻的记录。为了检索信息，计算机需要访问每一条记录，并通过其中的链接跳转到下一个记录，以此类推，直到找到用户所需的信息。如果用户想从一个庞大的数据库中检索出所有心脏病患者的病例，那么计算机就需要把每一条记录都访问一遍，才能将心脏病患者的病例筛选出来。这显然是一个非常缓慢而低效的信息检索方式，而且在那个年代，计算机和存储设备的

运转速度都慢得可怜，要执行这样的检索是难以想象的。或许这种事情在今天的人看来有些匪夷所思，不过第一代数据库的确是不能用来查找信息。

随着计算机和存储器的性能日渐提升，导航式数据库的局限性也愈发凸显。20世纪70年代，IBM的一名英国研究人员埃德加·弗兰克·科德（Edgar Frank "Ted" Codd）认为，数据库的模型亟待改进。科德与开发硬盘存储系统的IBM员工一道，设计了一个新的数据库模型，科德称之为关系模型⁽²⁵⁾。这种数据库结构比较智能化。数据的编排方式并不是像词典那样列出一长串条目，而是采用一系列数据库表⁽²⁶⁾，不同的数据库表通过关系键⁽²⁷⁾联系起来。

为了便于大家理解关系数据库，我们用词典来打个比方。关系数据库就好比一系列分类词典，比如动词词典、名词词典、副词词典等等。如果你想查找某个单词——比如动词“bow”，那就只需要翻阅动词词典。如果你想找“bow”的名词词义，那就只需要翻阅名词词典。这比从头到尾地翻阅一本超厚的大词典要高效多了。

关系数据库也是同样的道理，它采用关系键，将不同的记录⁽²⁸⁾联系起来。比方说，一个员工信息数据库可能会以员工编号作为关系键。知道了工号，你就可以在各个数据库表（好比各

个分类词典）中分别查询员工的工资信息、家庭住址、计算机登录情况等等。每个数据库表中的值都可以作为关系键，用于在其他数据库表中查找相应的信息。也就是说，员工的职位描述也可作为关系键，用来查询员工是否具备参与机密工作的资格。

20世纪70年代末出现了一种特殊的编程语言，用户可以通过它查询数据库中的复杂信息。这种语言脱胎于关系代数和微积分，全称为结构化英语查询语言（**SEQUEL**，即**Structured English Query Language**），后来又很快简称为结构化查询语言（**SQL**）。**SQL**支持查询、表达式、从句等多种编程元素。有了它，用户就可以在数据库中查询高度复杂的信息。比方说，你的公司建立了一个顾客信息数据库，要想在数据库中查询哪些顾客在2004年没有订购任何商品，那就可以使用这样的**SQL**查询语句：

```
SELECT customers.* FROM customers WHERE  
customers.customer_id NOT IN (SELECT  
customer_id FROM orders WHERE year  
(orders.order_date)=2004)
```

数据库管理系统负责接收和翻译**SQL**表达式，并尽快返回结果（如果情况理想的话）。**SQL**中包含好几种语言^{[\[29\]](#)}，其中数据操纵语言（**data manipulation language**，简称**DML**）和数据

定义语言（data definition language，简称DDL）能让用户运用简单易用的表达式在数据库中修改或添加数据。

此后，数据库在原有的基础上继续改进，开始向面向对象的趋势发展，并进一步得到优化。索引⁽³⁰⁾的概念得到引进，经常查询的信息存储在高速的临时数据库中，临时数据库则与主数据库相连。正如图书目录一样，索引可以使用户查找所需信息的速度大幅加快。如今，数据库的设计特点表现为高度模块化，可访问性强。可扩展标记语言（Extensible Markup Language，简称XML）、超级文本预处理语言（Hypertext Preprocessor，简称PHP）等基于网页的语言可以使用SQL，连接在线数据库。（有了网页服务器⁽³¹⁾，当你浏览的网页中包含这些高级语言写成的命令时，网页浏览器就会对它们进行翻译。）可以想见，未来数据库的分布将日趋广泛，可访问性日益增强，访问速度日益加快。

我们已经对数据库领域的巨大飞跃感同身受。就在几年前，家家户户的抽屉里还塞满了老照片，书架上堆满了书本、文件、录像带和CD，但是如今，我们查找、创建、购买的所有信息都是以数字形式保存下来的。大多数照片都已实现数字化，电子书正日渐占据图书市场的主流。大多数公司都喜欢给客户邮寄电子账单，而非纸质

账单。视频和音乐已大多实现数字化。多年来，我们的财富不过是存储在银行计算机中的数字。政府以数字化的形式发布信息；医院在计算机上保存病例。越来越多的商店开始实行网上交易。我们只需要轻轻点击几次鼠标，就可以购买食品、汽车乃至任何种类的商品。

所有这些海量的数据都存储在数据库中。它们的界面可以是一个赏心悦目的网页，供人随意浏览各种美观的照片；可以是一家网上商店，供人随心挑选品类丰富的商品；可以是一款功能强大的音乐播放程序，能让人徜徉于音乐的海洋；甚至可以是一款会计应用程序，能让人随时监测开支情况。但是，在多姿多彩的界面背后，让这一切成为可能的，便是精巧的数据库技术。

软件危机

如今，软件业可谓风生水起，蔚为大观，仿佛这个行业的发展向来一帆风顺，毫无波澜。但是，正如电子业历经了“数字暴政”的阵痛，才有了集成电路的诞生，新生的软件业也经历了危机的考验，这场危机还推动了一门全新学科的诞生。

程序员遇到的问题从一开始就很明晰。虽然有了新的高级编程语言的帮助，开发人员在编写代码的过程中可谓得心应手，但是，前所未有的

重大问题依然在不断涌现。20世纪60年代，随着集成电路的发明和摩尔定律的问世，计算机的性能每年都在突飞猛进。与此同时，软件应用也在以类似的速度不断飞跃。然而，就在这个过程中，程序员日益强烈地意识到，他们的程序正变得越来越难以掌控。程序当中的错误太多，软件并没有发挥应有的效果，而且开发系统的过程似乎正变得意料之外地漫长。

这些问题在1968年达到了顶点。在有史以来的第一次软件工程大会上，世界各地的计算机科学家齐聚一堂，共同讨论他们关切的问题。他们忧心忡忡，这一点从会上的讨论中就可以看出来：

麻省理工学院的罗伯特·格雷厄姆（Robert Graham）表示：“我们投入了长年累月的研究，耗费了巨大的投资，到头来却发现，我们从一开始就没有把系统研究透彻，软件根本没有取得预期的效果。我们就像莱特兄弟制造飞机一样，辛辛苦苦地把飞机造好，将它推下悬崖，任凭它轻而易举地坠毁，然后再从头开始。”

密歇根大学计算中心（University of Michigan Computing Center）的伯纳德·加勒（Bernard Galler）表示：“我想举几个在IBM碰到的不好的例子。有一次，用户提出，希望能够增强PL/1语言[\(32\)](#)可扩展性[\(33\)](#)。对此，IBM经过一周的内

部讨论，最终下结论称，这种事情不可能做到。因为语言设计师不打算告诉用户怎样实现所需的扩展。还有一个例子：OS/360⁽³⁴⁾操作系统的作业控制语言⁽³⁵⁾（job control language，简称JCL）开发出来以后，用户在设计阶段根本无法事先看到任何选项。为什么会出现这种情况呢？”

丹麦第一家计算机公司A/S Regnecentralen的员工彼得·诺尔（Peter Naur）表示：“……软件设计师的角色类似于建筑师和土木工程师，尤其是规划城市、工厂等复杂建筑的设计师。因此我们应该学会从这些领域吸取灵感，攻克我们遇到的设计问题。”

此次大会落幕后，另一场大会很快召开，其宗旨是讨论技术和管理思想。从这一刻开始，软件工程学作为一门新的学科登上了历史的舞台。伊恩·萨默维尔是现代软件工程师、英国圣安德鲁斯大学（St. Andrews University）教授，他撰写了很多业内的权威教科书。他表示，新学科的命名颇有内涵，意在表明，人们从此将采取系统的、有组织的方式来编写软件。不过，大会的主办方给出了不同的说法，“他们声称之所以发明这个术语，只是出于调侃的心态，没想到这种叫法就传播开了”。当然，这样命名是为了“故意制造煽动的效果”，以激发研究人员行动起来。

还别说，这样做真的起到了效果。没过多久，许多关键性的技术革新开始涌现，它们专门针对的是软件工程学领域，旨在辅助程序开发人员提高编程能力，写出高效的软件。在这一理念的指导下，新的编程语言得到开发。戴维·帕纳斯（David Parnas）等研究人员提出了信息隐藏的概念，这一概念在模块化编程和面向对象的编程领域举足轻重，它可以确保数据及相关函数封装在对象内，与其他的数据和函数分隔开来。这就好比采用标准化模块制造汽车——车载收音机等部件的更换不会影响到其他部件。同理，如果你设计好程序之后，突然想对某个地方进行改写（比方说将某个函数或数据结构更换），那么程序中的其他函数和数据完全不会受到影响。

集成开发环境（Integrated development environment，简称IDE）的发明，就是为了让编程变得更加轻松。它们的作用就好比文字处理器——程序开发人员可以在不同的窗口编写代码，并对其进行编译和调试。正如文字处理器可以检查拼写和语法错误，集成开发环境也可以帮助编程人员找出程序中的错误，并提供大量实用的工具进行除错。现代的集成开发环境通常包括一系列工具，比如：手持设备模拟器；用于设计图形用户界面（Graphical User Interface，简称GUI）的工具；全方位的帮助系统，用于辅助编程人员

从已有的库——即程序编程接口[\(36\)](#)

（application programming interface，简称API）中寻找合适的函数。

除了发明实用的编程工具，研究人员很快意识到，还有更好的软件设计方法亟待开发。这就好比汽车制造商不能把仅仅把目光放在硬件上，还应着眼于汽车的用途、目标客户的定位、成本开销的大小，从而将生产问题化整为零，分解成一个个具体的问题，比如：应该采用什么样的发动机、传动部件、转向系统、制动系统、车轮系统和座椅系统？车内需要容纳多少人？生产一台汽车需要多少时间？编写软件也是同样的道理。一个大型软件项目可能会比制造汽车复杂得多，怎样设计才能确保项目的高效运转？

研究人员很快意识到，要做到这一点，必须确立一个明确的软件生命周期[\(37\)](#)。首先，你必须合理定位产品和项目需求。接下来要做的，就是设计、运行和测试软件，并将其运行情况清晰地记录下来。最后要做的，就是发布软件，或许在这个阶段，你还需要指导用户如何高效地使用软件、如何进行必要的维护工作。

这些事情说起来容易，做起来难。产品或项目的需求并不好确定，因为顾客往往并不知道自己真正的需求到底是什么，因此可能会举棋不定、自相矛盾，甚至改变主意。他们往往不具备

编程人员的思维方式，因此不知道如何从软件开发的角
度表达自己的需求。换句话说，对于哪些事情在技术上可行，而哪些不可行，他们基本上没什么概念。有的时候，编程人员连目标用户是哪些人都无法确定，因此，真正应该提要求的人反而没有这个机会。

除此之外，不同的设计阶段应该如何开展，这也是摆在编程人员面前的一大难题。项目的开发架构是应该采用“瀑布模型^{[\(38\)](#)}”（waterfall model）——像流水下坡一样顺次开展每个阶段，还是应该采用“螺旋模型^{[\(39\)](#)}”（spiral model）——反复开展每个阶段，开发一系列原型软件，以最大限度地降低风险？软件的开发是应该遵循迭代式^{[\(40\)](#)}和增量式^{[\(41\)](#)}的过程，还是应该采用“灵活性强”的方式，以迅速适应可能发生的变化？就算你知道了开发软件的最佳顺序，怎样才能把每一个阶段的工作都做到最好？应该采用哪些设计方法和工具？怎样测试软件，才能确保万无一失？怎样维护软件，才能使之适应未来的变化？

正因为软件工程学专家孜孜不倦地攻克上述难题，软件项目的发展才得以适应硬件的发展需求和用户复杂的使用需求。软件工程学着眼于软件开发的方方面面，比如对软件体系结构进行建模，采用可视化程序设计，使用形式化方法^{[\(42\)](#)}

测试软件、提高其性能的可靠性。

伊恩·萨默维尔认为，软件工程学领域已经取得了许多重大的进步，这些进步给软件开发带来了实实在在的影响。“开发不同类型的软件，需要采用不同类型的软件技术和方法，”他表示，“配置管理^[43]（Configuration Management，简称CM）就是一个非常重要的项目管理方法，它支持并行开发^[44]。信息隐藏的概念是由帕纳斯在1972年提出的，它在抽象数据类型^[45]

（Abstract Data Type，简称ADT）领域得到了进一步发展，并影响了大多数现代编程语言。在关键任务系统^[46]领域，我们不仅可以采用安全分析和可靠性分析的方法，还可以运用容错技术^[47]，为航空器、化工厂等重要设施建立安全可靠的系统。统一建模语言（Unified Modelling Language，简称UML）将多种建模概念融合在一起，如今已成为软件系统建模的标准方式。编程环境的思想于20世纪70年代问世，在20世纪80年代得到进一步发展，如今已在软件工程学界得到广泛采纳。”

安东尼·芬克尔斯坦是软件工程学教授，在伦敦大学学院担任工程学系主任。他认为，正因为有了软件工程学的帮助，程序员才得以高效利用时间。“如果没有软件工程学，我认为硬件与软件之间的性能差距会进一步扩大。制约软件开

发的主要因素就是缺少训练有素的人才，现在也是如此。如果当初没有足够的人才投身于软件行业，我们就无法取得今天的成就了。”

不过，尽管为数众多的软件工程师竭尽全力地投身于技术攻坚，但是依然无法解决所有的问题。我们每次使用计算机时，都会对软件工程领域的一大问题感同身受：由于某种原因，软件每次升完级以后，其运转速度似乎都会变慢。计算机科学家（及众多编程语言的发明者）尼古拉斯·维尔特（Niklaus Wirth）观察到了这一现象，人们将其称为维尔特定律（Wirth's Law）。维尔特定律的内容是：软件变慢的速度永远快过硬件变快的速度。其他科学家也发表过类似的观点。曾在英特尔担任研究人员的兰德尔·肯尼迪

（Randall Kennedy）就是其中一人。他曾写道：“虽然与几年前的Office 2000相比，Vista系统上的微软的Office 2007虽然处理能力提高了将近两倍，但是占用空间却多出了11倍以上。”造成这一现象的罪魁祸首是软件膨胀（software bloat）——新版本的软件往往只是在原版本的基础上叠加了新的代码，而并没有经过重新编写。维尔特定律表明，纵使计算机的运行速度快得惊人，新一代的软件的运行速度也比不上十年前的老版本。

目前，计算机科学家正在费尽心思解决这个

问题。不过萨默维尔深知，这并不是软件工程领域的唯一挑战。“软件项目面临的主要挑战在于，开发环境正变得越来越复杂。这是因为，我们在建立新系统的过程中，将不同的供应商提供的各种系统和服务整合了起来。原本，在软件工程领域，很多理论之所以能够成立，往往是基于这样的前提，那就是，系统完全处在软件开发人员的掌控之下，软件开发人员可以做出明智的决定来开发和改变系统。当这样的前提不再成立时，软件测试等方法必须做出相应的调整，以适应新的情况。”

这一观点引起了一些计算机科学家的高度重视。伦敦大学学院教授马克·哈曼围绕搜索问题对软件工程进行了研究。他按照遗传算法——也就是由“优胜劣汰”的生物进化规律演化而来的随机化搜索方法，利用计算机来寻找特定软件的最优测试方法。

“软件测试是衡量软件质量、寻找优化方法的关键手段之一。测试的内容之一，就是寻找特定的输入，使程序执行特定的代码片段，”哈曼表示，“人工完成这样的工作需要花费很大的心血。这就好比你得从厚厚的地址簿里大海捞针，将一个人的电话和住址找出来。但是，如果计算机能够对测试用例[\(48\)](#)进行评定，我们可以轻而易举地将这个过程自动化。”

不过，仅仅有个聪明的测试方法是不够的。我们所说的危机与20世纪60年代末的软件危机已经无法同日而语，但是从某种程度上讲，软件业依然面临着定位问题。大型软件开发项目依然经常失败，尽管软件工程师尽了最大的努力。有时候，问题只是出现在成本和时间上——我们并不擅长估算软件开发产生的耗费。“我们的软件估算水平很差，”芬克尔斯坦表示，“如果你让我‘开发一个售卖二手教科书的网页前端⁽⁴⁹⁾，’我可能一时半会儿没法告诉你需要多少时间。运气好的话，我可能之前就做过一次这样的项目，但是我在这方面的经验也就仅限于此了。”

不过，重大的失败也有可能是由更微妙的原因造成的（而且这样的事情不在少数，造成的损失也极为惨重）。芬克尔斯坦认为，问题的症结并不在于分析师和开发者经验不足。“个人认为，这些系统之所以失败，都是一些人们很熟悉的问题造成的：比如在项目设计和开展阶段犯了低级错误，”他表示，“你必须问自己，既然人们对这些问题都很熟悉，为什么还要一次又一次地犯同样的错误？他们是傻子吗？他们没看过萨默维尔的书吗？这些问题前十页就讲了，哪怕稍微翻翻也好啊，又不一定要全读完！我觉得，问题的症结是，企业结构、管理决策结构与开发软件的技术过程存在不协调。因此，这些系统之所以

会失败，问题出在管理上，而不是工程上。”

软件工程领域的挑战还表现在，怎样满足客户的信息安全需求，怎样设计软件才能使之适应互联网时代的潮流。此外，一些所谓的“非功能性需求”也越来越值得重视，比如电源或电池寿命。在这个问题上，哈曼的想法比较现实：“如果我在乘坐越洋航班时，笔记本的电量在半路上就耗光了，那么就算里面的软件再好，对我来说一点用也没有。我宁愿软件耗电量低点，让续航时间久点，就算软件里头到处都是漏洞也没有关系。”芬克尔斯坦也这么认为：“在软件工程领域，我们一直致力于开发新特性，从来没有想过省电的问题——一般都不会有人往这方面想，除非你要设计航天器。但是现在，这个问题已经变得很关键。有很多之前没有考虑过的新特性一下子变得非常重要。”

还记得上一章我们讲到的万福玛利亚传球吗？那是帕特森给整个行业的科学家提出的挑战。芬克尔斯坦确信，它会给软件工程领域带来重大的变革。“软件工程领域面临的另一大挑战就是多核技术。如果软件无法充分发挥多核技术的优势，那么一台计算机就算内核再多也于事无补。要是采用一个内核就足以运行所有软件，那么剩下15个内核留着干嘛呢？总不可能都用来运行杀毒软件吧。我们所需要的，不仅仅是编程技

术的一个突破，而是整个软件工程领域的革新。这肯定会改变我们当前的游戏规则。”

虚拟技术的未来

软件从未变得如此重要。当年赴摩尔学院听计算机讲座的先驱们万万没有想到，他们所学的知识竟然催生了一个新的行业，一个由计算机指令支撑起来的虚拟行业。软件分为多个层次，涉及到微代码、机器代码、汇编语言、高级语言的转换。数据库、软件开发过程，乃至整个企业结构都与之息息相关。计算机无所不在，它们的思想统领着整个世界。软件的重要性已变得无以复加，无怪乎世界上财力最雄厚的富豪当中，有好几个人是靠经营软件公司而积累了大笔的财富。

“优化软件就相当于优化一切。”

注释

(1) 所谓扰频，就是指改变频率使（电话等信号）不被窃听。

(2) 达德利是英国的一座城市，位于英格兰西米德兰兹郡。

(3) 我们习惯把那些不使用电源，电路里只有一个半导体元件的收音机统称为“矿石收音机”。矿石收音机是指用天线、地线以及基本调谐回路和矿石做检波器而组成的没有放大电路的无源收音机，它是最简单的无线电接收装置。

(4) 即剑桥大学的物理系。

(5) 法国知名的玩具品牌，专业生产金属拼装玩具。

(6) 美国哈佛大学应用数学教授霍华德·阿肯受巴贝奇思想启发，在1937年得到美国海军部的经费支持，开始设计“马克一号”（由IBM承建），这是美国第一台大尺度自动数位计算机。此后马克二号、马克三号和马克四号相继问世。

(7) 数学实验室即今天的计算机实验室，也就是剑桥大学的计算机科学系。

(8) 存储程序计算机最早是由著名数学家冯·诺伊曼等人在1946年总结并明确提出来的，因此又被称为冯·诺伊曼计算机。存储程序计算机在体系结构上主要特点有：以运算单元为中心；采用存储程序原理；存储器是按地址访问、线性编址的空间；控制流由指令流产生；指令由操作码和地址码组成；数据以二进制编码。

(9) 工作记忆是一种对信息进行暂时加工和贮存的能量有限的记忆系统。

(10) 水银延迟线存储器就是作为最早的计算机的内存（主存器）来使用的。因为当时没有任何现成的器件可以直接拿来使用。为了寻找更好的存储器，人们费尽了心血，几乎所有能利用的物理现象，电、光、声、磁都被探索过来。研制埃尼阿克的工程师莫奇利想到了水银延迟线（Mercury Delay Line）——二战期间为军用雷达开发的一种存储装置，作为最早的计算机上的内存。

(11) 旋风计算机（Whirlwind），一款由麻省理工大学研制的早期电子计算机。引入了当时先进的实

时处理理念，并最先采用显示器作为输出设备，与以往机械系统的电子置换不同，拥有世界首款成熟的操作系统。其设计理念直接被美国空军的SAGE防空系统继承，并对20世纪60年代的商用计算机产生巨大影响。

(12) 在计算机指令系统的优化发展过程中，出现过两个截然不同的优化方向：复杂指令集和精简指令集。用通俗的话来说，复杂指令集就是将更多功能步骤集成在了CPU中。例如烹饪西红柿鸡蛋的步骤包括：购买西红柿和鸡蛋，清洗食材，打鸡蛋，炒鸡蛋，放西红柿继续炒，放盐，炒熟之后盛盘上桌。这样已连续的步骤被集成在CPU内部。外部程序需要烹饪西红柿鸡蛋的时候只需要向CPU下达指令说：“烹饪西红柿鸡蛋”就可以了。精简指令集就是在CPU中只有容器，调料等基本指令。需要人们在软件中写上如何协调这些指令的步骤，例如你需要向CPU下达指令说：买西红柿，清洗食材，打鸡蛋，炒鸡蛋，放西红柿继续炒，放盐，炒熟之后盛盘上桌。

(13) 函数（function）是一种子程序，利用函数名称，可以接收回传值。函数与程序在某些编程语言中是被分开的，但是在C语言中，所有程序都是函数，这两者被认为是相同的。

(14) 程序框图是算法的一种，又叫流程图，是一种用规定的程序框、流程线及文字说明来准确、直观地表示算法的图形。在程序框图中，圆角长方形表示起、止框，平行四边形表示输入、输出框，长方形表示处理框、执行框，用于赋值、计算，菱形表示判断框，成立写是或Y，不成立则写否或N。

(15) 控制流的英文是control flow，它是程序执

行中所有可能的事件顺序的一个抽象表示。

(16) 模块化设计，简单地说就是程序的编写不是开始就逐条录入计算机语句和指令，而是首先用主程序、子程序、子过程等框架把软件的主要结构和流程描述出来，并定义和调试好各个框架之间的输入、输出链接关系。逐步求精的结果是得到一系列以功能块为单位的算法描述。以功能块为单位进行程序设计，实现其求解算法的方法称为模块化。模块化的目的是为了降低程序复杂度，使程序设计、调试和维护等操作简单化。

(17) 把汇编语言书写的程序翻译成与之等价的机器语言程序的翻译程序称为汇编程序。汇编程序输入的是用汇编语言书写的源程序，输出的是用机器语言表示的目标程序。

(18) 助记符（英语：Assembly Mnemonic）通常可在以汇编语言写成的电脑程序内找到。它是一种帮助我们人写程序的代码，并不会真的写到程序里去运行，它是一个语法或是保留字，功能是用来帮助程序设置一些低级的设置，如程序起始地址、变量的存储器地址等东西。

(19) 在计算机科学和软件工程领域，形式化方法是基于数学的特种技术，适合于软件和硬件系统的描述、开发和验证。

(20) 面向对象是一种对现实世界理解和抽象的方法，是计算机编程技术发展到现在一定阶段后的产物。早期的计算机编程是基于面向过程的方法，例如实现算术运算 $1+1+2=4$ ，通过设计一个算法就可以解决当时的问题。随着计算机技术的不断提高，计算机被用于解决越来越复杂的问题。通过面向对象的方式，

将现实世界的物抽象成对象，现实世界中的关系抽象成类、继承，帮助人们实现对现实世界的抽象与数字建模。通过面向对象的方法，更利于用人理解的方式对复杂系统进行分析、设计与编程。同时，面向对象能有效提高编程的效率，通过封装技术，消息机制可以像搭积木的一样快速开发出一个全新的系统。面向对象是指一种程序设计范型，同时也是一种程序开发的方法。对象指的是类的集合。它将对象作为程序的基本单元，将程序和数据封装其中，以提高软件的重用性、灵活性和扩展性。

(21) 封装就是将抽象得到的数据和行为（或功能）相结合，形成一个有机的整体，也就是将数据与操作数据的源代码进行有机的结合，形成“类”，其中数据和函数都是类的成员。封装的目的是增强安全性和简化编程，使用者不必了解具体的实现细节，而只是要通过外部接口，以特定的访问权限来使用类的成员。

(22) 简单地说，指针就是指向变量和对象的地址。为了访问内存，需要对内存里的单元进行编号如0、1、2、3.....这的确很像房子跟房子的编号，比如南新大街7栋（起始地址），401房（偏移地址），为了存放这些编号（地址），就用了一个变量，即指针变量（它的值可能是0或1或2.....它指向的就是一个具体的存储单元）。

(23) 链表是一种常见的重要的数据结构。链表有一个“头指针”变量，以head表示，它存放一个地址。该地址指向一个元素。链表中每一个元素称为“结点”，每个结点都应包括两个部分：一为用户需要用的实际数据，二为下一个结点的地址。因此，head指向

第一个元素：第一个元素又指向第二个元素；……，直到最后一个元素，该元素不再指向其它元素，它称为“表尾”，它的地址部分放一个“NULL”（表示“空地址”），链表到此结束。

(24) 用有向图结构表示实体类型及实体间联系的数据结构模型称为网状模型。

(25) 关系数据库是采用关系模型作为数据组织方式的数据库。关系数据库的特点在于它将每个具有相同属性的数据独立地存储在一个表中。对任一表而言，用户可以新增、删除和修改表中的数据，而不会影响表中的其他数据。关系数据库产品一问世，就以其简单清晰的概念，易懂易学的数据库语言，深受广大用户喜爱。

(26) 在关系数据库中，数据库表是一系列二维数组的集合，用来代表和储存数据对象之间的关系。它由纵向的列和横向的行组成，例如一个有关作者信息的名为authors的表中，每个列包含的是所有作者的某个特定类型的信息，比如“姓氏”，而每行则包含了某个特定作者的所有信息：姓、名、住址等等。

(27) 关系键是关系数据库的重要组成部分。关系键是一个表中的一个或几个属性，用来标识该表的每一行或与另一个表产生联系。

(28) 表中的一行称为一个记录。一个记录的内容是描述一类事物中的一个具体事物的一组数据，如一个雇员的编号、姓名、工资数目，一次商品交易过程中的订单编号、商品名称、客户名称、单价、数量等。

(29) 结构化查询语言SQL包含6个部分：数据查询语言（DQL: Data Query Language）；数据操作语言

(DML: Data Manipulation Language)；事务处理语言 (TPL)；数据控制语言 (DCL)；数据定义语言 (DDL)；指针控制语言 (CCL)。

(30) 索引是对数据库表中一列或多列的值进行排序的一种结构，使用索引可快速访问数据库表中的特定信息。

(31) 网页服务器是指驻留于因特网上某种类型计算机的程序。当网页浏览器（客户端）连到服务器上并请求文件时，服务器将处理该请求并将文件反馈到该浏览器上，附带的信息会告诉浏览器如何查看该文件（即文件类型）。服务器使用HTTP（超文本传输协议）与客户机浏览器进行信息交流，这就是人们常把它们称为HTTP服务器的原因。网页服务器不仅能够存储信息，还能在用户通过网页浏览器提供的信息的基础上运行脚本和程序。

(32) PL/1，是Programming Language One的简写。它是一种IBM公司在1950年代发明的第三代高级编程语言，用于IBM的MVS、或迪吉多的VAX/VMS等操作系统中。在系统软件，图像，仿真，文字处理，网络，商业软件等领域均可应用。有些类似PASCAL语言。

(33) 可扩展性是指扩充或增强开发环境功能的能力，即增加一些原来没有的功能。可扩展性对于开发环境来说并不是新的概念。有些语言开发工具包就提供了这样的能力，它们允许为其用户开发环境增加功能。

(34) 1964年，IBM发布了S/360系统，对应的最早的主机操作系统为OS/360，命名360的意思是指一个圆的360度角，表示全方位的服务。OS/360支持多道程

序，最多可同时运行15道程序。为了便于管理，OS/360把中央存储器划分为多个（最多15个）分区，每个程序在一个分区中运行。

〔35〕 作业控制语言（job control language简称JCL）是用于描述MVS，OS/390和VSE操作系统上描述作业（执行单元）的语言。描述操作系统对全部作业分配计算机的运行时间和资源。作业是分步骤进行的，在编制程序时需要加入特殊语句使作业能够分步进行。

〔36〕 应用程序编程接口是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件的以访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。

〔37〕 软件生命周期（SDLC，Systems Development Life Cycle, SDLC）是软件的产生直到报废的生命周期，周期内有问题定义、可行性分析、总体描述、系统设计、编码、调试和测试、验收与运行、维护升级到废弃等阶段，这种按时间分程的思想方法是软件工程中的一种思想原则，即按部就班、逐步推进，每个阶段都要有定义、工作、审查、形成文档以供交流或备查，以提高软件的质量。

〔38〕 瀑布模型是一个项目开发架构，开发过程是通过设计一系列阶段顺序展开的，从系统需求分析开始直到产品发布和维护，每个阶段都会产生循环反馈，因此，如果有信息未被覆盖或者发现了问题，那么最好“返回”上一个阶段并进行适当的修改，项目开发进程从一个阶段“流动”到下一个阶段，这也是瀑布模型名称的由来。

〔39〕 螺旋模型基本做法是在“瀑布模型”的每一

个开发阶段前引入一个非常严格的风险识别、风险分析和风险控制，它把软件项目分解成一个个小项目。每个小项目都标识一个或多个主要风险，直到所有的主要风险因素都被确定。

(40) 迭代式开发是一种与传统的瀑布式开发相反的软件开发过程，它弥补了传统开发方式中的一些弱点，具有更高的成功率和生产率。在迭代式开发方法中，整个开发工作被组织为一系列的短小的、固定长度（如3周）的小项目，被称为一系列的迭代。每一次迭代都包括了需求分析、设计、实现与测试。采用这种方法，开发工作可以在需求被完整地确定之前启动，并在一次迭代中完成系统的一部分功能或业务逻辑的开发工作。再通过客户的反馈来细化需求，并开始新一轮的迭代。

(41) 增量模型融合了瀑布模型的基本成分（重复应用）和原型实现的迭代特征，该模型采用随着日程时间的进展而交错的线性序列，每一个线性序列产生软件的一个可发布的“增量”。当使用增量模型时，第1个增量往往是核心的产品，即第1个增量实现了基本的需求，但很多补充的特征还没有发布。客户对每一个增量的使用和评估都作为下一个增量发布的新特征和功能，这个过程在每一个增量发布后不断重复，直到产生了最终的完善产品。

(42) 形式化方法是基于计算机的系统的软件开发的需求、规格、设计、编程和实现的数学方法。

(43) 配置管理是通过技术或行政手段对软件产品及其开发过程和生命周期进行控制、规范的一系列措施。配置管理的目标是记录软件产品的演化过程，确保软件开发者在软件生命周期中各个阶段都能得到

精确的产品配置。

(44) 因开发和维护的原因，要求能够实现开发人员同时也在同一个软件模块上工作，同时对同一个代码部分作不同的修改，即使是跨地域分布的开发团队也能互不干扰，协同工作，而又不失去控制。这就叫并行开发。

(45) 抽象数据类型是指一个数学模型以及定义在该模型上的一组操作。抽象数据类型的定义取决于它的一组逻辑特性，而与其在计算机内部如何表示和实现无关。即不论其内部结构如何变化，只要它的数学特性不变，都不影响其外部的使用。抽象数据类型可以使更容易描述现实世界。例：用线性表描述学生成绩表，用树或图描述遗传关系。

(46) 如果一个系统的可靠性对于一个组织是至关重要的，那么该系统就是该企业的关键任务系统。

(47) 容错就是当由于种种原因在系统中出现了数据、文件损坏或丢失时，系统能够自动将这些损坏或丢失的文件和数据恢复到发生事故以前的状态，使系统能够连续正常运行的一种技术。

(48) 测试用例（Test Case）是为某个特殊目标而编制的一组测试输入、执行条件以及预期结果，以便测试某个程序路径或核实是否满足某个特定需求。

(49) 用互联网来做比喻，凡是通过浏览器到用户端计算机的统称为前端技术。相反，存贮于服务器端的统称为后端技术。

响彻世界的声音

吼猴⁽¹⁾是世界上叫声最响的动物之一。即使在厚厚的密林中，它们震耳欲聋的吼声依然可以传到几公里以外的地方。人类虽然没有这么夸张的嗓音，但是在当今时代，有了计算机的帮助，就算是最轻柔的低语浅唱，也可以传遍整个世界。

计算机是喜好社交的机器，它们在横越高山大海、遍布广袤大陆的网络中一刻不停地互相交流。它们的语言举世通用，不分国别；它们的脉动承载着人类的工业、知识、文化、思想乃至愿望。

知识的虚拟网络将人们空前紧密地联系起来，它给人们带来自由的同时也带来了新的问题：是不是每一个与我们交流的人都值得信赖？我们是否需要新的方式来保护隐私？

克劳德·香农把手从棋子上移开，抬头看着坐在对面的年轻对手，犀利的目光咄咄逼人。“将军！”

时值1955年，香农尚在贝尔实验室工作，不过这几天，他因公事在身，需要在帕罗奥多市短暂停留。酷爱下棋的他，时不时就会跑到帕罗奥多国际象棋俱乐部（Palo Alto Chess Club），找人切磋棋艺。五、六年前，香农发表了好几篇关于计算机下国际象棋的论文，他对计算机的智能化发展越来越感兴趣。这天晚上，与香农对弈的棋手是斯坦福大学（Stanford University）的22岁

博士生彼得·柯尔斯坦（Peter Kirstein），这场对弈显然让他乐在其中。柯尔斯坦很早就开始下国际象棋，要想赢他实属不易。他并不熟悉香农，只是觉得眼前这位39岁的科学家很有魄力，虽然身形极为单薄，但是眼神犀利如剑。

此次对弈是两名计算机先驱人物的偶遇，尽管当事人对此浑然不知。香农已经功成名就；柯尔斯坦则是未来的新星，他将在香农的基础上取得新的成就，这样的成就是两个人都未曾预料的。

尽管香农的一项研究成果给日后的柯尔斯坦带来了巨大的影响，但在1955年，这项成果对于香农而言已是过眼云烟。它的初衷是为了消除电话通信中的噪声干扰。当时，贝尔实验室的研发团队一直在着力攻克长途通信的问题。那时候，电话通信采用的是模拟信号^{[\(2\)](#)}。听筒中的送话器将声音转化为电信号：音调高的声音信号转化为高频电信号（表现在波形图上，就是每秒钟产生的波峰和波谷数量更多），音调低的声音信号转化为低频电信号（表现在波形图上，就是每秒钟产生的波峰和波谷数量更少）；响度大的声音信号转化为振幅大的电信号（即波峰更高，波谷更低），响度小的声音信号转化为振幅小的电信号（表现为平缓的波形）。如果你在长线路中传输这样的模拟电信号，那么信号就会在传输的过

程中衰减，因为长线路的电阻大。解决这个问题
的方法就是采用放大器，由它接收微弱的模拟信
号，将信号的强度放大，并传送出去。然而，不
幸的是，线路越长，信号的频率和振幅就越容易
发生畸变而失真，噪音的干扰也越严重。多级放
大器的串联在放大信号强度的同时，也会引起失
真和噪音的叠加，从而使信号的失真越来越大，
噪音的问题越来越严重。

1948年，香农决定将他的布尔逻辑思想应用
到电话网络上。他认为，电话通信不应该采用模
拟信号，因为容易受到噪音的干扰。要想避免噪
音的干扰，只需要采用两种类型的信号——即1
和0。这是最简单的信息类型（如果信号只有唯
一的取值，那就没什么意义了，因为它表达的信
息量太少）。香农将这个只有1和0两种状态的信
息类型称为二进制数位，或比特。他表示，只要
将所有的模拟信息都简化为比特，那么，要把它
们从背景噪音中分离出来就会容易许多。这就好
比在一个嘈杂的派对上，如果你的朋友想跟你谈
话，你根本就听不清楚他在说什么，因为周围的
杂音太大了；但如果他只是对着你大叫，你完全
可以从周围的杂音中分辨出他的叫声。香农的思
想催生了数字通信技术，其原理就是将音频信号
转化为二进制数位（一连串的比特），二进制数
位携带了关于音频波形的信息。音频信号在呼出

端转化为数字信号，以1和0组成的比特序列传播，并在接收端还原成模拟音频信号。这样一来，信息在传播过程中就能与噪音清晰地分隔开来，不受任何干扰。真是绝妙的方法。

香农又花了几个月的时间将所有的理论难点理清楚，完成这项工作后，他并未止步于此，而是进一步将二进制数位在通信领域的所有应用潜力都阐述了出来。他提出，可以使用不同频率的信号来传输比特，也就是说，同一时间可以在同一条线路上传输多个比特（好比音调各不相同的人同时在派对上大声呼喊）。他揭示了信息传输速率的极限，还阐述了信息传输的纠错方法——只要运用这些方法，就算有些信号在背景噪音中无法分辨，原信号也依然可以在毫无损失的情况下重建出来。此外，香农提出了压缩的概念——即利用更少的二进制数字携带更多信息的方法，还给出了信息率、信道容量、平均信息量的精确计算方法。这些理论具有重要的学术价值。香农以此为基础，撰写了一篇论文——《通信的数学理论》（A Mathematical Theory of Communication）。《通信的数学理论》为一个全新的研究领域奠定了基础，这个研究领域称为信息论⁽³⁾。

香农的思想造成了深远的影响。可能读者朋友们已经注意到了，凡是人们可能使用到的任何

信息处理设备——比如计算机、电话、电视、收音机、MP3播放器——都是数字设备。它们使用二进制代码，即0和1的语言。在当今时代，几乎所有信息的传输、存储和操控都是以数字为载体，并不是采用原始模拟信号的形式。无论是文字、图片、照片、音频还是视频，这一点概莫能外。“显然，在过去50年里，通信速度之所以一次又一次得到提升，大多是因为人们对香农的成果进行了更灵活的应用，无论是建立调制解调器连接还是铺设100 GBPS光纤，”通信学教授乔恩·克劳克罗夫特表示，“其中涉及到的研究领域无非就是信息论、编码技术和调制技术⁽⁴⁾。”可以说，香农的成果推动了数字革命的兴起。

1949年，香农和他在贝尔实验室的同事贝蒂（Betty）结婚。两人育有三子。1956年，香农进入麻省理工学院，并在那里度过了下半辈子的职业生涯。他虽然一生成就卓著，但却平易近人，安稳顾家，而且为人很有幽默感，兴趣广泛得惊人。

香农发明了许多五花八门的器件。他手工制造了三十多台独轮车，这些独轮车有的配上了发动机，有的安装着稀奇古怪的偏心轮，骑起来颠簸不堪，还有的型号太小，根本骑不了。除此之外，他还制造了一台处理罗马数字的计算机、好几个会玩杂耍球的机器人，还实践了早期的人工

智能理念，研制了好几台会下国际象棋的计算机。

克劳德·香农在晚年患上了阿尔茨海默氏症（Alzheimer's disease，即所谓的老年痴呆）。可惜的是，他虽然推动了信息革命的兴起，却始终未能充分享受到信息革命带来的便利。2001年，香农在疗养院逝世，享年84岁。

架设计算机的桥梁

当初和香农对弈时，彼得·柯尔斯坦对香农的研究工作知之甚少。他正在斯坦福大学攻读博士，主攻微波[\(5\)](#)技术，与研究电话网络的人没什么联系。这种情况日后将发生改变。

1933年，柯尔斯坦在德国出生。当时，希特勒已经掌权五个月左右，犹太人的日子变得越来越不好过了。1935年，犹太人开始公开受到迫害，因此，柯尔斯坦的父母决定移民英国。柯尔斯坦的母亲顺利拿到了签证，因为她刚好出生在英国（她父母曾经在英国工作过几个月，她就是在那时候出生的）。1936年，柯尔斯坦的母亲移居到了英国，1937年，柯尔斯坦父子俩也紧随其后。这次移民无疑拯救了全家人的性命。

移民后不久，小柯尔斯坦的父母为了让他改掉吮手指的坏毛病，骗他说，吮手指的小孩子在英国会被警察叔叔切掉大拇指。没过多久，他们

就发现小彼得竟然站在一大堆警察面前吮手指，他还一边吮，一边（用德语）问警察：“你们的剪刀呢？”由此可见，柯尔斯坦在很小的时候就已经是一个实验主义者了。

柯尔斯坦在战时经历了三次逃难，八岁时开始在伦敦的海格特学校（Highgate School）上学。他喜欢下国际象棋、集邮、学数学、听古典音乐。凡是感兴趣的学科，他都学得很好，除此之外的学科都学得一塌糊涂。他在学校非常活跃，喜欢打网球。升上预科班后，他决定以后学习数学专业，于是开始专攻数学。他很聪明，提前两年考完了O-level⁽⁶⁾和A-level⁽⁷⁾（只不过，他后来为了拿剑桥大学的奖学金，又去考了一次A-level）。上完中学后，学生一般都要服几个月的兵役，但是柯尔斯坦靠着亲戚的关系逃掉了兵役，跑到加州大学洛杉矶分校（University of California, LA，简称UCLA）待了一个学期（半年），后来又找了份暑期工，给美国军队打杂。一年后，他又回到美国，继续做这份暑期工。他的工作是做一台“人体计算机”，给机场计算降雨径流⁽⁸⁾面积。

接着，柯尔斯坦进入剑桥大学，学习数学专业。入学第二年，到了选择专业方向的时候，他开始考虑要不要进入数学实验室，还和实验室主任莫里斯·威尔克斯谈了谈——当时，威尔克斯的

EDSAC 2计算机已经研制完毕，投入使用。不过，经过深思熟虑，他还是决定在大三这年学习电气工程专业。毕业后，柯尔斯坦获得了斯坦福大学的奖学金，赴美国攻读理科硕士和博士学位。他的博士学业一开始进展并不顺利——导师换了好几个。但是，一旦所有的事情都回到正轨，柯尔斯坦的学业就开始进展神速，他在短短六个月的时间里就拿到了博士学位，又花了一年的时间讲授课程。为了从繁重的工作中稍事喘息，顺便体验一下滑雪的乐趣，柯尔斯坦来到了瑞士新建成的欧洲核子研究组织⁽⁹⁾（法语：Organisation Européenne pour la Recherche Nucléaire；英语：European Organisation for Nuclear Research，通常被简称为CERN）。他在那里待了四年的时间，接触到了一些世界上最新、最快的计算机。

柯尔斯坦从欧洲核子研究组织离职后，收到了好几所高校伸来的橄榄枝，但他选择了留在瑞士（有部分原因是为了滑雪），供职于美国的通用电气（General Electric，简称GE）。这个决定影响了他未来的人生轨迹。柯尔斯坦在通用电气的工作职责是为公司追踪相关科研领域的最新进展。这给了他绝佳的机会遍访各地高校，广结精英人脉。除了英语和德语，他还熟练掌握了法语、俄语等各种外语。在接下来的30年时间里，

他虽然有别的工作，但在通用电气的工作从未间断。这一过程使他对整个计算机科学领域形成了独到的视角和全方位的了解。

1965年在伦敦出差期间，他表示有兴趣加入伦敦大学（University of London）的计算机科学研究所（Institute of Computer Science）。他认为，在计算机之间建立远程连接是一个非常具有实际意义的做法。他选择了两台计算机，一台是当时英国最快的计算机——卢瑟福·阿普尔顿实验室（Rutherford Appleton Laboratory）研制的IBM 360—75，另一台是一个图形终端，称为IBM 2250图形显示单元（IBM 2250 Graphics Display Unit）。伦敦大学的计算机科学研究所没有科研经费可以支持这个项目，因此，柯尔斯坦自己争取了经费，开始作为讲师在研究所工作，直到1967年。他雇佣了一个名叫史蒂夫·威尔伯（Steve Wilbur）的学生共同参与这个项目，在威尔伯的协助下，项目取得了很大的成功。

将计算机连成“网络”的思想正日渐成为研究领域的热门话题。早在20世纪60年代早期，约瑟夫·利克莱德（Joseph Licklider，人称JCR或“Lick”）等美国计算机科学家就提出，全球的计算机网络终有一天或许会取代图书馆。1965年，英国国家物理实验室（National Physical Laboratory，简称NPL）的计算机科学家唐纳德·

戴维斯（Donald Davies）提出，可以以分组交换技术为基础，建立一个覆盖全国的数据网络。所谓分组交换，就是将用户传送的数据划分成一定的长度，每个部分叫做一个分组，通过传输分组的方式传输信息的一种技术。

柯尔斯坦所研究的，恰恰是这个新兴的网络领域。几乎同一时期，他在加州大学洛杉矶分校面见了一位朋友，朋友给他介绍了一位名叫文特瑟夫（Vint Cerf）的研究生。（文特后来成为了互联网之父）。他们似乎在开展一个新的项目，将多台计算机连接起来。柯尔斯坦觉得这些思想非常有趣。当时，利克莱德设法说服了他在国防部高级研究计划局（Advanced Research Projects Agency，简称ARPA）的同事鲍勃·泰勒（Bob Taylor），使之相信，建立计算机网络是一个有价值的项目。泰勒雇佣了麻省理工学院的研究生拉里·罗伯茨（Larry Roberts）来监管这个项目。不久后，高级研究计划局委托BBN技术公司

（BBN Technologies）制造新的计算机，用于在网络中传输信息。阿帕网（ARPANET）于1969年末问世——起初，接入阿帕网的机构还比较少，只有加州大学洛杉矶分校、斯坦福研究所（Stanford Research Institute）、犹他州立大学（University of Utah）和加州大学圣巴巴拉分校（University of California, Santa Barbara）。人们

视之为一个实验性质的网络，如果不是因为全国、乃至全世界的研究人员对它的兴趣越来越大，项目甚至有可能就此终止。不久后，其他高校乃至美国政府也开始接入阿帕网，阿帕网中的计算机数量每个月都在缓步增长。

1970年末，美国高级研究计划局的拉里·罗伯茨向英国国家物理实验室的唐纳德·戴维斯提出，两个机构的网络可以互连（当时，英国国家物理实验室已经建立起了内部分组交换网络）。他们意识到，美国核监测研究室（US Nuclear Monitoring Research Office）在阿拉斯加州、蒙大拿州和挪威的三个地震台阵^{[\[10\]](#)}建立的连接是可以利用的现成资源。网络信号可以通过海底电缆传输到伦敦，再经由贡希利卫星地球站

（Goonhilly Satellite Earth Station）传输到人造卫星，通过人造卫星传输到华盛顿市，最后传输到阿帕网。罗伯茨由此提出，可以利用这种方式，实现国家物理实验室与阿帕网的互连。但是，欧美当时的政治形势不允许英国政府的实验室和美国建立如此紧密的联系。戴维斯无法接受罗伯茨的提议。他只能另找金主——一个能提供5000英镑资金支持的金主。这样一来，合适的人选便只剩下一个人，那就是彼得·柯尔斯坦。

对于国际网络，柯尔斯坦有着自己的想法（1968年，他主持了国家物理实验室的一次会

议，讨论连网问题——40年后，他又担当了同样的角色，主持了一项年度活动）。他建议，可以利用伦敦大学计算机科学研究所和卢瑟福·阿普尔顿实验室之间已经建立起来的连接，将英国最大的计算机用作阿帕网的一个远程主机。他还提出，应该让其他英国高校也能使用这个网络。然而，政治形势依然对他不利，而且，连剑桥大学都决定不支持这个项目——莫里斯·威尔克斯出于经济原因做出了这个决定。

尽管困难重重，但柯尔斯坦绝不是一个轻言放弃的人。他或许可以称得上是世界上人脉最广的研究人员了，熟人到处都是。那时候，英国邮局掌管着电话线路，他在邮局认识好几位领导，很快就争取到了免费连网一年的优厚待遇。尽管在当时，要从美国的BBN技术公司进口必要的设备并不容易，但是他想尽办法做到了这一点。1973年7月25日，伦敦连上了阿帕网。虽然连网的计算机一共只有30台左右，不过阿帕网已经是名副其实的国际网络了。

多元网络的兼容并包

尽管阿帕网横越了大西洋的屏障，成为了国际性的网络，但是，就不同网络的计算机之间如何沟通的问题，人们尚未达成共识。形形色色的计算机网络在越来越多的国家生根发芽。每一个

网络内部通行的语言都与其他网络有所不同，这使得不同的网络之间变得互不兼容。为了将欧洲的网络与阿帕网相连，柯尔斯坦不得不开发协议转换[\(11\)](#)器。“我别无选择，只能将截然不同的网络连接进来，”他说。

这时候，莫里斯·威尔克斯正在研究自己的网络解决方案——剑桥环网（Cambridge Ring）。威尔克斯曾对柯尔斯坦说：“我们之间有一个很大的不同——我可以随心所欲地做自己想做的网络，而你却得时刻迁就别人的标准和协议[\(12\)](#)！”

“但是不迁就别人也不行，”柯尔斯坦表示，“虽然剑桥环网在技术上非常先进，但是威尔克斯从来没有将它标准化，所以剑桥环网从来没有普及开来。”

不能让截然不同的网络永远泛滥下去。1972年，罗伯特·卡恩（Robert Kahn）开始供职于高级研究计划局的信息处理技术办公室（Information Processing Techniques Office，简称IPTO）。他意识到，必须为阿帕网开发出更好的方法，以最大限度地减少信息传输过程中出现的错误。他雇佣了文特·瑟夫做助手。1974年，他们发布了一种新的“连接语言”，也就是便于不同的计算机之间互相沟通的互联网协议。他们的方法在今天被称为互联网协议套件，其实，更通行的叫法是

TCP/IP，它是“传输控制协议和互联网协议”（Transmission Control Protocol and the Internet Protocol）的英文缩写。不过，这样的叫法稍微有些简略了，因为计算机网络体系结构的通信协议至少可以划分为四个不同的层次。

为了便于大家理解互联网协议的概念，我来给大家打个比方。假设有两家大企业的总裁想要共享某些信息。第一家企业的老板决定跟第二家企业的老板商谈，给她发送一份文件。当然，他身份地位高，手头事务忙，不可能亲力亲为，于是就命令手下的通信总监打理细节问题。通信总监办事情非常彻底。首先，他联系了对方公司的通信总监，确认了接收事宜。接着，他仔细地将文件切分成细小的信息块，对每个信息块都附上了标签，注明了信息内容和接收人。然后，他将信息块连同标签一起送到公司的收发室，将每一个信息块精心打包，将地址标注在包裹上，等待邮递员派送。邮递员将好几个包裹收集起来，放在箱子里妥善保管，然后确定将邮件派送到对方企业收发室的最佳路径。如果他在中途损坏了某些包裹，或者发现它们的包装不够严实，就会折返回来要求更换。对方企业收到包裹后，会将它们转交给通信总监。通信总监则根据标签，细心地将零散的信息块重新整合成完整的文件。如果发现有的信息块在派送的过程中不幸丢失或损

坏，她就会要求第一家企业重新发送相应的信息块。等到所有的信息块都顺利送到、组装到位后，她就会将整合好的文件提交给自己的老板。

计算机网络通信或多或少也是这个道理。上面所说的老板就相当于计算机的应用程序，比如网页浏览器和网页服务器；通信总监相当于TCP（传输控制协议）层；收发室相当于IP（互联网协议）层；邮递员相当于地址层，也就是低层次的传输方法，比如以太网（Ethernet）、IEEE 802等局域网协议，以及点对点协议（Point-to-Point Protocol，简称PPP）等框架协议。

TCP/IP是一个设计精巧的封装或抽象⁽¹³⁾系统，它能够确保信息在电话线、卫星链路、光纤电缆上准确无误地传输，同时，应用程序或TCP/IP数据的使用者也无须了解信息传输的具体机制。这也就是为什么人们常说，TCP/IP只需要两个空罐和一根线⁽¹⁴⁾就能实现信息传输了。

鲍勃·卡恩和文特·瑟夫的发明可以说是计算机科学领域的一个重大突破，这项成果只要运用得当，就可以使计算机网络通信更加可靠，更容易实现。但是设计新的标准协议是一回事，将它普及开来就是另一回事了。这么多年来，依然有新的网络类型在不断涌现。在英国，自从所谓的“彩色书”协议（"coloured book" protocol，得名于协议规范文档的封面颜色）问世以后，新的协

议层出不穷（尽管柯尔斯坦大力倡导采用可兼容的方法）。它们为了追求特立独行的效果，已经到了不择手段的地步。连计算机的地址都被彩色书协议颠倒了过来，格式从原来的“peter@cs.ucl.ac.uk”改成了“peter@uk.ac.ucl.cs”。

柯尔斯坦仍然在想方设法将所有的网络连接起来，尽管它们互不兼容。这样一来，用户就不会因为协议冲突而无法与其他网络互连。不过柯尔斯坦后来对自己的做法产生了怀疑。“我到现在还不知道自己到底是在做好事还是在做坏事，我感觉自己在变相纵容英国人随心所欲地开发网络协议，反正无论开发出什么，都可以与其他网络互连。”

1978年，柯尔斯坦和瑟夫发表了一篇文章，讨论不兼容网络的连接问题，这是世界上第一篇关于此类话题的论文。（这时候，瑟夫已经是柯尔斯坦在国防部高级研究计划局的项目负责人。）文中提出了两种截然相反的观点。瑟夫认为，网络协议只要有一种就够了，没有必要开发那么多。但是柯尔斯坦觉得应该把视野放宽一些，他深知，现在世界上存在许多不同类型的网络。“我并不认为有哪种协议能够做到一家独大，”柯尔斯坦表示，“不过我必须承认，文特一直是个有远见的人。他认为TCP/IP协议能够通行

全球。所以我们合写了论文，把两种观点都收录了进去，无论事情朝哪个方向发展，这篇论文都能给人们提供一定的参考。”

伦敦大学的计算机科学研究所于1973年关闭，因此，柯尔斯坦的团队加入了伦敦大学学院的统计与计算机科学系，1975年开始在系里办公（他们带来了为数众多的设备，伦敦大学学院的数学家为此大开眼界）。1976年，柯尔斯坦与英国政府的多个部门签订了合同。此后，他又拿到了英国国防部的合同，负责深入开发英美之间的计算机网络。除此之外，他还参与了一些国际性的研究项目，研究卫星通信和无线电通信。在众多有识之士的努力之下，英国越来越多的地方开始连网。

1980年，伦敦大学学院计算机科学系创立，柯尔斯坦虽然不太想当系主任，不过还是接下了这个职位。他一共在任了15年，直到他的老学生史蒂夫·威尔伯取而代之。在此期间，他的团队招纳了好些重要的网络研究人员，其中包括乔恩·克劳克罗夫特和马克·汉德利。

柯尔斯坦继续致力于连接不可兼容的网络，确保它们能够支持TCP/IP。“每当新的网络技术问世，我们都会确保它能够得到有效利用，并且从一开始就与互联网整合。许多从事网络技术的人根本不知道、也不关心互联网的存在。”

经过艰苦卓绝的努力，文特·瑟夫的预测变成了现实。1981年，TCP/IP成为了阿帕网的标准协议。一年后，在柯尔斯坦的推动下，伦敦大学学院的跨大西洋卫星链路将原来的通信协议改成了TCP/IP。欧洲核子研究组织建立了基于TCP/IP协议的网络，澳大利亚等国也纷纷效仿。在美国，网络研究的发展呈加速趋势，国家航空航天局（National Aeronautics and Space Agency，简称NASA）、国家科学基金会（National Science Foundation，简称NSF）和联邦政府能源部（Department of Energy，简称DOE）都建立了基于TCP/IP协议的网络。

1990年，阿帕网光荣退役。从1991年起，英国联合科研网（Joint Academic Network，简称JANET，是英国大学学术和研究的计算机系统骨干网络）开始支持IP协议。在接下来十年里，新的网络纷纷将原来的协议改成TCP/IP。至此，世界上大多数网络终于开始使用同一种语言：互联网的语言。

正如我们所知，没过多久，一种类似于互联网的新兴事物诞生了，那就是由众多网络连接在一起而形成的规模惊人的新网络。它的规模每年都在扩大，至今仍然如此。正如生物学家专注于研究大脑的神经网络，计算机科学家也致力于研究日益庞大的计算机网络。周实（Shi Zhou，伦

敦大学学院教授)就是这样一名研究人员。他认为,互联网之所以能够保持高效运转,是因为具备一些非常重要的特性。为了便于大家理解,我先给大家讲解一下周实接下来会提到的一些重要概念⁽¹⁵⁾。所谓节点,通俗地说,就是指某个大环境中的一个点。在计算机领域,每一个工作站、服务器、终端设备、网络设备,即拥有自己唯一网络地址的设备,都可以称为网络节点。整个网络就是由这许许多多的网络节点组成的。所谓节点的度,就是指直接和某个节点相连的节点数目;一般认为,节点的度越大,该节点所代表的个体就越居于网络的中心,占有的网络资源越多。网络中度大的核心节点为“富节点”(rich node)。“互联网的一大重要特性在于,它表现出异配连接倾向(disassortative mixing),”周实表示,“也就是说,度小的边缘节点倾向于和度大的核心节点(也就是富节点)相连。互联网还表现出所谓的“富人俱乐部现象”(rich-club phenomenon),也就是说,少数富节点紧密地连接在一起,形成了一个“俱乐部”,将网络资源集中起来,在互联网中发挥了中枢的作用,能够为信息传输提供大量可选的捷径。因此,互联网中典型的信息传送路径可以分为三步,第一步是从源节点⁽¹⁶⁾传输到富人俱乐部中的一个富节点,第二步是在富人俱乐部中的两个节点之间传输,

最后一步是从富节点传送到目的节点[\(17\)](#)。”这些重要的特性是随着互联网规模的扩大而自然产生的，没有经过任何人为的计划和设计。

互联网沟通了整个世界，它的地位举足轻重。互联网之所以能够诞生，不仅仅是因为技术上的进步，而且还因为一些有识之士当年克服了政治因素的阻碍，力排众议，推进了不同国家之间网络连接的进程。乔恩·克劳克罗夫特对此深有感触（如今他已是剑桥大学计算机实验室的马可尼通信学教授[\(18\)](#)）。他表示，柯尔斯坦极大地推动了互联网的兴起，将网络“互联”的特性发展到了极致。“可以说，彼得·柯尔斯坦造就了互联网，”克劳克罗夫特表示，“因为他使原本互不兼容的网络兼容并包，使互联网成为了最广泛的网络之网。”

“这话说得不假，”柯尔斯坦笑了笑，说道，“文特认为，全世界的网络都可以统一起来，人们只需要使用一种网络协议。我不这么认为。我觉得，我们还有很长的路要走。文特的想法在美国可以实现，但在英国不行，英国总有人开发不同的网络协议。因此，必须有人发挥桥梁的作用，使各种类型的网络兼容并包。我们必须考虑到，不同的国家和地区有着不同的发展步伐，这不仅仅表现在技术上，也表现在政治上。要想推动互联网的发展，必须因地制宜。”

柯尔斯坦具有独到的全球视角和无可匹敌的广阔人脉，正因为如此，他一次又一次地推动了欧美国家乃至全世界范围的技术合作。当被问及此生最大的成就时，柯尔斯坦回答道：“我最引以为豪的事情，就是一次又一次地架设了国际合作的桥梁。”互联网的诞生既是技术推动的结果，也是政治努力的结果，这一点并不奇怪，毕竟，互联网的影响范围遍及全世界。“我觉得的确是这样，”柯尔斯坦表示，“互联网的诞生确实是离不开技术的进步，但前提是，人们得对它有信心才行。我在这方面算是尽了自己的一份力量。”

瑟夫是彼得·柯尔斯坦四十多年的老朋友和老同事。他说：“在互联网的发展演进过程中，伦敦大学学院发挥了中心作用。”

柯尔斯坦（如图6）至今还在伦敦大学学院工作。他致力于帮助中亚和非洲地区的发展中国家开发和改善联网能力，连接互联网。伦敦大学学院的网络研究团队依然领先世界。随着全球性的互联网日益成熟，怎样塑造它的未来，也成为了至关重要的问题。而柯尔斯坦目前所做的，正是解决这个问题。



图6 彼得·柯尔斯坦在伦敦大学学院计算机科学系机房，他身后的架子上摆满了现代计算机服务器。2011年摄。

网络编址，编织成功的未来

从网络发展的早期阶段开始，人们就意识到，计算机需要有地址，这样才能确定信息的发送方和接收方，从而确保正常通信。编址的方案之一，就是给计算机分配号码（正如电话公司给我们分配电话号码一样），然后将编号存储在主目录中。但是这样做有很多问题：号码对人来说很难记忆，使用英文单词会更方便一些；况且，如果所有编号都保存在一个目录中，那么同时查询地址的请求就会太多，导致运行速度缓慢；更糟糕的是，一旦目录出错，整个网络系统就会瘫痪。

阿帕网等大多数网络都采用了两种类型的地

址。一种是由英文字母或名称组成，另一种是由数字组成。人们平常书写时使用的是便于识记的英文字符，机器收到字符后，将其转换为对应的数字，这样便能知道信息接收方的精确地址。互联网也采用了这套方法，并且沿袭至今。我们现在看到的网站域名就是互联网的编址方案之一（与之对应的另一套方案就是IP地址）。

国际互联网工程任务组（The Internet Engineering Task Force，简称IETF）是全球互联网最具权威的技术标准化组织，主要任务是负责互联网相关技术规范的研发和制定。伦敦大学学院计算机科学系教授马克·汉德利就是这个组织的成员。他表示，用户在互联网浏览器中输入特定域名时，计算机会采取好几个步骤进行域名解析。他给我们通俗地讲解了其中的过程。“假设我在地址栏中输入的是bbc.co.uk，”他说，“首先我们必须明确的是，bbc.co.uk只是一个域名。要想访问网站，浏览器必须知道与之对应的数字化IP地址才行。”

“将域名转换为IP地址的过程称为域名解析，域名解析需要专门的服务器，称为域名服务器（domain name server，以下简称DNS）。我在浏览器中输入域名后，计算机就会询问本地DNS，‘bbc.co.uk的IP地址是什么？’”

“本地DNS可能也不知道答案，不过它知道

全世界仅有的13台互联网根服务器[\(19\)](#)的IP地址。于是，它就会去找其中一台根服务器，‘我想知道bbc.co.uk的IP地址。’”

“根服务器一般不会存储具体的域名，所以它也不知道bbc.co.uk的IP，但它知道uk这个顶级域[\(20\)](#)的服务器IP，于是，它就会回复说，‘这些是uk域名的服务器地址，你找它们问问吧。’”

“于是，本地DNS就会询问其中一个uk域名服务器，‘bbc.co.uk在哪里？’，对方就会回答，‘我不知道bbc在哪里，但我知道co域名的服务器地址，你找它们问问吧。’”

“接下来，本地DNS向co域名服务器发送请求后，终于找到了bbc域名的服务器地址。它向对方询问，‘bbc.co.uk在哪儿？’对方就会把IP地址报给它，于是它就会把结果报给我的计算机。”

“之后，我的网页浏览器就会开始连接bbc的服务器。本地DNS会发出建立TCP连接的请求，bbc的服务器接受请求后，连接就建立起来了。我的计算机会向对方索取文件，而对方则会直接把文件发送到我的浏览器上。”

这样的域名解析方式看起来似乎太过复杂，但是它能确保整个网络系统始终保持极为稳定的状态，不会轻易瘫痪。没有一台服务器掌握着所有计算机的IP地址，但是很多服务器都掌握着部分计算机的地址。因此，就算互联网的重要服务

器瘫痪，也不会影响到普通用户。毕竟，不管有多少台服务器瘫痪，总会有其他服务器可以派上用场。

汉德利对此深有体会。2011年3月，日本发生大地震，造成超过10,000人伤亡，大部分地区受灾严重，满目疮痍。“发生了这么大的地震，互联网却几乎没怎么受到影响，”他说，“从亚洲进出的互联网流量只是略微有所下跌，下跌的幅度只有5%左右，而且没过多久就恢复正常了。互联网有足够的冗余⁽²¹⁾和足够的鲁棒性。一条线路出了问题，总是可以换别的线路。”

在阿帕网时期，有一个专门的机构负责分配域名和IP地址，该机构称为互联网号码分配局（Internet Assigned Numbers Authority，以下简称IANA）。IANA主要是乔恩·伯斯塔尔（Jon Postel）一个人在发挥作用。如今，它已被互联网名称与数字地址分配机构（Internet Corporation for Assigned Names and Numbers，简称ICANN）接管，域名分配权下放给了各个注册人和公司，所以域名可以购买和出售，不过IP地址的分配权依然由ICANN控制。

2011年初，已经没有IP地址可以分配了，所有的地址都分配完了。之所以会出现这种情况，是因为当前我们采用的是互联网协议版本4（Internet Protocol version 4，简称IPv4）的地址

系统，IPv4的地址长度只有32位，分为4段，每段8位，用十六进制数字表示，每段数字的范围为0~255，段与段之间用句点隔开，例如159.226.1.1。这就意味着地址总数大约只有4,294,967,296个，其中还有许多留作特殊用途。43亿的数字听起来很大，但实际上根本供不应求。这给互联网的未来造成了很大的威胁。

为了解决这个问题，一个常用的方法是网络地址转换（Network Address Translator，简称NAT），它能使整个子网的所有主机只通过一个公有IP地址访问互联网。这样做虽然有投机取巧之嫌，但不失为一种便利的手段，因此得到了广泛运用。不过，由于地址转换后的主机并没有建立真正的端对端连接，很多应用程序在使用时会出现问题。

汉德利早在多年前就预见到了IP地址终将用完的状况。“早在90年代，这个问题就已经变得非常明显，所以我们提出了好几种解决方案，用来替代现有的地址系统。互联网工程任务组提出了一个标准化的系统，称为互联网协议6（IP version 6，以下简称IPv6）。IPv4采用的是32位地址长度，而IPv6采用的是128位地址长度。可以说，有了IPv6，地球上每一粒原子都能分配到一个IP地址。”

或许说来有些奇怪，尽管IP短缺的重大问题

已经有了切实可行的解决办法，整个互联网依然在沿用IPv4的地址系统。对此，汉德利举了个类似的例子。“1967年，瑞典把道路通行方向从左行改成了右行。为了这次改动，瑞典已经计划了很多年。所有的道路交叉点都改成了适合双向行驶。最终，瑞典在某个升旗日[\(22\)](#)将左行改成了右行。互联网的更新换代也是同样的道理，从IPv4升级到IPv6，是一个事关重大的调整，需要周密的计划。”

互联网经历的最近一次变革是在1981年，当时，由于阿帕网正式采用了TCP/IP协议，全世界大约有400台计算机需要做出相应的改动。如今，一旦IPv6得到推行，需要改动的计算机将达到几十亿台之多，而且这一数字每天都在以几千台的速度增长，因为联网的手机和平板电脑越来越普及。要想在一夜之间改变整个互联网的编址系统，已经是天方夜谭。况且，人们也没什么动力做出改变。和当年互联网问世时一样，IPv6的推进，不仅需要工程上的努力，还需要政治上的努力。在这种情况下，文特·瑟夫和彼得·柯尔斯坦站到了一条阵线上，也就不足为奇了。两人都是大力倡导编址系统升级的先行者。

“我觉得，随着IP地址渐渐用完，越来越多的人会注意到这个问题，”瑟夫表示，“但问题是，真要到那个时候，一切都晚了。我觉得，在危机

迫近时再采取行动不是明智之举。所以我一直大力倡导将IPv6和IPv4结合起来使用，越快越好。”

“这个问题不是光靠一张嘴皮子就能解决的。我们要做的，不仅仅是说服互联网服务提供商采用IPv6，”柯尔斯坦表示，“我们还得试着改变这个行业，改变互联网的业务流程，改变人们的思维方式。我们得从业务流程着手，寻找推动合作的方式。”

万维网的扩张

尽管计算机可以通过互联网提供的网络和“通信语言”互相沟通，不过它们沟通的方式很快就被一项新的事物所支配。我们称之为万维网（World Wide Web）。这项改变世界的简单发明恰恰是一名先行者的灵感结晶。

蒂姆·伯纳斯-李（Tim Berners-Lee）1955年出生于伦敦。他的父母都是数学家，两人是在研制“费伦蒂·马克一号（Ferranti Mark I）”计算机的过程中认识的。蒂姆从小就开始接触计算机输出设备（五孔纸带）。或许在此之前，没有哪个孩子是在这样的环境中长大的。正因为成长环境特殊，小蒂姆自然而然喜欢上了电子学和数学。长大后，他选择了攻读物理专业，平常喜欢在业余时间研制自己的计算机，还在一家电信公司找了份工作。

1980年，25岁的伯纳斯-李成为了欧洲核子研究组织的合同研究员。当时，欧洲核子研究组织建立了欧洲最大的计算机网络，在这里工作的科学家经常需要分享信息和文件。伯纳斯-李提出，可以设计一个新的程序，帮助人们查找信息。新程序的设计围绕着一个中心概念，称为超文本 [\(23\)](#)。伯纳斯-李并不知道前人在这方面的研究成果。他编写的程序可以将数据库中的信息放在页面中展示，页面中有许多词语可以“链接到”数据库中的其他页面。选定其中一个词语后，程序会将相关的页面调取出来。通过这种方式，用户可以快速、直观地访问大量的信息。伯纳斯-李将这个新程序称为ENQUIRE。

1984年，伯纳斯-李作为正式员工回到欧洲核子研究组织时，他当年设计的程序还在使用。不过，他意识到，尽管ENQUIRE非常实用，但这个集中式的数据库维护起来非常费事。“显然，人们需要一个既像ENQUIRE那样实用、又对所有人开放的工具，”他说，“我想将ENQUIRE的规模扩大，这样一来，如果有两个人一开始分别使用了ENQUIRE，后来又在这个平台上协作，他们就可以将各自的信息链接起来，而不需要做任何其他的改动。这就是Web的概念。”

1989年，伯纳斯-李和他在欧洲核子研究组织的同事建立了一个简单的Web系统，用于处理电

话号码。这个Web系统分为两个部分：一是“服务器”，它由一台计算机充当，在收到请求后会运行程序，提供信息；二是“浏览器”，它们分别在一台或多台计算机上运行，会根据用户的指令向服务器发出请求，然后将收到的信息以易于读取的形式展现给用户。两年后，伯纳斯-李通过互联网，将服务器和浏览器软件向所有人开放，然后在网上进行了一些宣传。后来他回忆道：“从那时候开始，我收到了很多对Web感兴趣的互联网用户发来的反馈，他们的想法、灵感、源代码以及精神支持让我受益匪浅。这些都是我在身边的环境中难以得到的。可以说，是互联网的用户造就了Web。”

很快，在伯纳斯-李的号召下，世界各地的编程人员纷纷开始编写浏览器，用于访问早期Web网页。其中涌现了不少杰出的作品。赫尔辛基理工大学（Helsinki University of Technology）的学生开发了可在X-Windows系统上运行的Erwise浏览器，加州大学伯克利分校的学生编写了ViolaWWW浏览器。康纳尔法学院（Cornell Law School）设计了Cello浏览器，这是适用于微软早期Windows系统的第一款浏览器。伊利诺伊大学厄巴纳-香槟分校（University of Illinois at Urbana Champaign）的国家超级计算应用中心（National Center for Supercomputing Applications）发布了

Mosaic浏览器，它很快就成为了一款多平台浏览器。

鼠标 / 图形用户界面风行开来，Web服务器的数量呈现出指数式的增长。互联网上的Web流量突然一路飙涨。“总流量在1993年3月还只有0.1%，9月份就增长到了1.0%，12月份就已经达到了2.5%，Web的增长速度在互联网上是空前的。”伯纳斯-李表示。到了2010年，互联网上至少有75%的流量来自Web。其中四分之一的流量来自Web网页，Web视频成为了新的流量来源，在正常的电视收视时间贡献了37%的流量（平均每分钟上传的新视频总时长达到了35小时。）

伯纳斯-李为万维网的诞生立下了三大不可磨灭的功劳。第一个重要贡献在于，他提出了一个重要的思想：每个文档或图片都应该有自己的标识符，正如每台计算机在互联网中都有自己的标识符。伯纳斯-李在与互联网工程任务组合作期间，决定制定一个与UNIX目录结构类似的标准，采用正斜杠“/”将文件和目录分隔开。举个例子，如果有一个Web文件，文件名为“welcome.html”，所属目录（或文件夹）为“news”，运行Web文件的服务器地址是“bbc.co.uk”，那么，它的统一资源定位符（Universal Resource Locator，简称URL，俗称网页地址）就可以写成“www.bbc.co.uk/news/welcome.html”。

这也就是为什么我们如今访问特定的网页时都需要输入URL地址。URL表明了文件、目录的精确位置和计算机的地址。用户在Web网页上看到的所有文件，都是由浏览器从URL地址上传上来的——网页上的图片、文本、广告很有可能来自不同的地方。它们都有自己的URL地址，浏览器可以据此将它们一一下载，并展示给用户。

伯纳斯-李的第二个重要贡献在于，他发明了一种特殊的高级语言，称为超文本标记语言

（hypertext markup language，简称HTML）。它可以告诉浏览器如何显示Web网页中的复杂内容，使用起来简易灵活，任何想设计Web网页的人都可以轻松上手。此外，HTML适用于广泛的平台，任何HTML文件都可以通过Web服务器迅速、轻易地传输到任何类型的计算机平台上，再由该平台的Web浏览器进行翻译，从而将编写者想要实现的一切效果——文本、图像、超链接等各项内容都在网页中显示出来。为了方便用户使用，HTML只采用文本和一系列代码（或标签），写出来的文件非常简洁，下面举一个例子：

```
<html>

<head>

  <title> My Webpage about the Internet </title>

</head>

<body>

  <h1>The Origins of the Internet</h1>

  <p>Here' s a diagram of the first four nodes of the ARPANET:</p>

  <img src = 'http://www.peterjbentley.com/1969ARPANETimage.gif '

    width = '520' height = '421' >

  <p>This a hyperlink to <a href = ' http://www.example.com ' > the website
www.example.com </a>.</p>

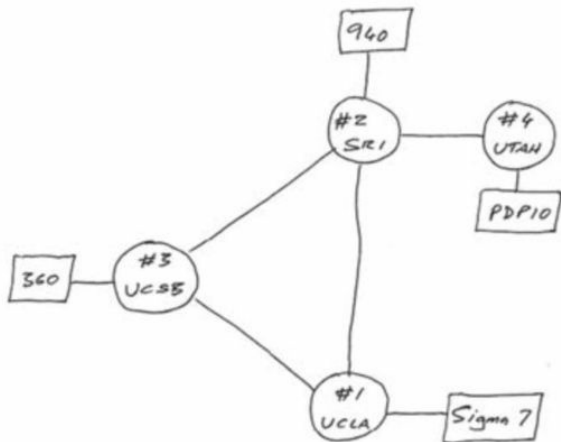
</body>

</html>
```

以下是该文件经过Web浏览器翻译后形成的网页效果图：

The Origins of the Internet

Here's a diagram of the first four nodes of the ARPANET:



This a hyperlink to [the website www.example.com](http://www.example.com).

从图中可以看出，网页的标题显示在了浏览器窗口的标题栏里。对比HTML文件和这张简洁明了的效果图，我们不难看出网页中的其他内容在HTML文件中是如何定义的。带超链接的文本采用了下划线标注，颜色也与其他文本有所不同。如今，HTML语言已经升级到了4.0以上的版

本，任何使用计算机的人都能体会到，Web网页可以变得多么实用、复杂而精美。

伯纳斯-李的第三大贡献在于，他发明了超文本传送协议（Hypertext Transfer Protocol，简称HTTP），HTTP协议可以将HTML文件从Web服务器传送到Web浏览器。这种语言并不是TCP/IP的替代协议，而是基于TCP/IP的一个应用层协议

[\[24\]](#)。不知道大家还记不记得我们之前打过一个比方，讲的是两个总裁想要共享信息的例子，在这个例子当中，TCP层相当于通信总监；IP层相当于收发室。如果把HTTP也放进去，它就相当于一个统一的指令集，两名总裁可以使用这种特殊的语言互相索要资料，这些资料也就相当于万维网中的文本、图片和视频。两名总裁想要传达的信息还是像从前一样，通过TCP/IP层分割成块，分开传输。

两台计算机建立连接后，HTTP协议假定其中一台是客户端，另一台是服务器。客户端（比如Web浏览器）采用HTTP语言，向服务器发送请求，索要特定的资源，比如一个HTML文档或一张图片。服务器采用同样的语言返回特定的信息（比如资源的大小、保存时间以及细节信息，以便客户端查询本地的缓存）。如果客户端的本地缓存中没有该纪录，则服务器会将客户端索要的资源发送过去。HTTP协议中定义了好几种命令

（也叫“动词”），以方便客户端和服务端互相发送信息。如果你的浏览器需要读取数据，就会向服务器发送“显示”（GET）命令。如果网页上有一张表格，你填好之后，点击“发送”，浏览器就会连同你刚才输入的数据一起，向服务器发送“投递”（POST）命令。

现在的万维网已经能够使服务器和客户端之间采用非常高级的通信形式。程序可以运行，数据库可以检索。用户可以在一些特殊的网页上创建和编辑内容，这些网页统称为维基（wiki）。社交网站（比如Facebook和LinkedIn）、博客等新事物广受欢迎，它们正彻底革新着人与人之间的交流方式。Web视频成为了新时代的电视机。不过，纵使网络设计的创新千变万化，精巧复杂，它们的根基无非是一些简单质朴的概念，那就是URL地址、HTML语言和HTTP协议。

1994年，伯纳斯-李转职到麻省理工学院，并在麻省理工学院创立了万维网联盟（World Wide Web Consortium，又叫做W3C），该机构旨在为万维网未来的发展保驾护航。2004年，伯纳斯-李获颁爵士勋衔。他被英国南安普敦大学

（Southampton University）聘为教授，负责一个新的研究项目：开发语义网^{[\(25\)](#)}（Semantic Web）。2009年，他成为英国政府的顾问。伯纳斯-李还创立了万维网基金会（World Wide Web

Foundation），该机构旨在推进Web在技术和社会领域的发展，并致力于保持万维网的开放性和无偿性。

“Web就像一张白纸，它不会把自己的用途明明白白地写在纸上。你得自己开发它的用途，善于利用它来处理日常生活中的所有信息流。我的首要任务就是确保Web技术的发展符合人的长远利益。”蒂姆·伯纳斯-李表示。”

编织错综复杂的网络

如今，除了娱乐休闲，人们还在互联网上传输重要信息，打理电子商务，甚至参与选举。如果网络安全没有保障，这一切都不可能实现。网络安全理论的起步，离不开两名先行者——克劳德·香农和艾伦·图灵取得的突破性成果。

从二战期间到战后，为了帮助政府更加安全地传输信息，香农和图灵研究了密码学。他们的大多数研究成果都是机密级文件，从未公诸于世，不过香农还是发布了一些自己的成果。

尼古拉斯·库尔图瓦是伦敦大学学院的计算机科学家，他的专业领域是密码学。他认为，香农的工作意义重大。“香农不仅是通信学和信息论之父，还是密码学之父。他在20世纪40年代发表了两篇关于通信学和密码学的论文。直到20世纪70年代，这两篇论文还是密码学领域仅有的文

献。他把密码学从一门艺术变成了一门科学。他的成果为密码学日后所有的发展埋下了种子。”

要想保证网络安全，方法主要有三种。每一种都比上一种更先进，更安全。第一种方法就是，一不做二不休，把软件设计得无比复杂，让人根本弄不懂它的工作机制——将内部算法完全保密。库尔图瓦表示，这种方法最基本，也最常用。“这就好比你在做一件人人都琢磨不透的事情。这种方法之所以有效，是因为没有人知道你到底在做什么，这样一来，你所做的事情就是完全保密的。有些人把它称为‘隐晦式安全’（security by obscurity）。直到现在，商业活动大多采取这样的原则。不过，这种方法很早以前就被破解了，它并不能很好地保障网络安全，因为黑客往往可以将软件破解。只要给他们足够的时间和咖啡，他们就能侵入系统。”

二战期间，“隐晦式保密”的效果并不好。政府很快意识到，他们需要更安全的通信方法，让敌人破解不了。因此，图灵、香农这样的编码者和破译者在战时开发出了第二代安全保密方法——采用密钥保证信息安全。发信人使用密钥将信息加密，相当于把信息锁在盒子里，将盒子（加密信息）发给收信人，收信人使用同样的密钥才可以解锁（破译）。就算有人把锁的构造（加密方法）琢磨透了，如果没有密钥，照样打

不开。采用这种方法可以使需要保密的信息量大大减少。“你可以把加密算法、源代码乃至密码系统的任何细节都公开，”库尔图瓦说，“只要把一小部分关键的信息——也就是密钥保护起来就可以了。”

如今，这套对称密钥加密算法已经广为人知。我们平时给自己的计算机和银行账号设置密码，就是采用了这种方法。密码就相当于密钥。我们将计算机和银行账号用密码“锁住”，只有输入相同的密码才能“解锁”。但是，采用这种方法的问题在于，你得把密钥与收信人共享。显然，你不能将密钥通过互联网传输，以免密钥在传输过程中被人监控和拦截。（如果你想对密钥本身加密，就得再共享一个密钥来解密它）。因此，为了保险起见，重量级信息的密钥或许得通过快递实体传输，或者当面递交。此外，采用这种方法还会带来另一个问题：需要创建的密码太多。

很多人的账户密码太多，到最后连自己都不记得了。你去给公司安全审计就会发现，很多人在计算机和显示器周围贴满了便笺——这一般就是他们记录密码的地方。柯尔斯坦谈到这个问题时笑了起来：“很多人为了确保账户安全，坚持勤换密码，结果密码太多，根本记不住，只好专门找一个地方写下来。但是，写出来贴在显眼位置的密码还叫密码吗？”

只要找到密码，拦截传输中的密钥，或者通过破解加密算法拿到密钥，就可以破译通过该密钥加密的任何信息。如果你把所有账户的密码设成同一个，那就麻烦大了。后果很严重。“很多公司的安全系统都非常薄弱，”库尔图瓦说，“他们使用很多张智能卡⁽²⁶⁾，每张卡的密码都一样。只要破解了一张卡的密码，所有的卡都可以随便使用了。”

库尔图瓦深知这种加密算法在黑客面前多么不堪一击，因为他自己就是行家，为了做研究，经常破解密码。“我已经成了破译专家，破译是我的工作，”他说，“破译涉及到数学和加密工程学领域的知识，不过这些知识也可能被别有用心的人利用！我已经成功破解了不少加密系统，其中有些还是符合国际安全标准的政务系统和商务系统。”库尔图瓦破解的系统包括Oyster智能卡（伦敦交通卡，可用于乘坐地铁等）和全世界各种通达卡的加密系统。他之所以能发明自己的破译方法，也是受到了香农的直接启发。他在一次讲座中了解到了香农的理论。“香农发表过一些成果，但是这些成果近50年来一直没有引起重视。他曾经在论文中说，破解复杂密码的过程，实质上就是对一个代数方程系统求解的过程。这也是他定义一个加密算法是否安全的标准。近50年来，没有人能从中找到密码破译的方法。我是

最早证明出这种代数数学方法可以破解加密系统的人之一。”

对称密钥加密算法显然并不完美，但在相当长的一段时间里，没人能提出更好的方案。人们一度认为，这就是终极的加密算法了。此后，算法和密钥虽然能够得到改进（事实上也确有改进），但终究跨越不了分享密钥的这道大坎。但是，正当人们放弃希望之际，新的加密算法出乎意料地问世了。这个方法最早由英国政府通信总部的研究人员于1973年发明，但被列为绝密，不能发表。直到三年后，美国的研究人员惠特菲尔德·迪菲（Whitfield Diffie）和马丁·赫尔曼

（Martin Hellman）做出了同样的成果，新的方法才公诸于世。两年后，麻省理工学院研究人员里维斯特（Ronald Rivest）、沙米尔（Adi Shamir）和阿德勒曼（Len Adleman）以他们的姓氏首字母命名了一个比较完善的密码体系，称为RSA体系。

第三代加密算法称为非对称密钥加密（又叫做公开密钥加密）。它的数学原理极为精妙，但又非常简单。所谓非对称密钥，是指一对加密密钥与解密密钥。你可以将加密密钥公开，解密密钥私人存放，这样一来，每个人都可以用公开的加密密钥给你发送信息，但是这些加密信息只能通过你的私人密钥解密。这就好比你把一个自动

上锁的盒子交给别人，自己保留钥匙。人人都可以往盒子里放信息，但只有你的密钥才可以打开盒子。你不需要再和别人共享私钥，也不需要再对密钥本身加密。

非对称密钥加密算法的提出，是数学史上的一个奇迹，每个人都对此深感意外。库尔图瓦一提起它就变得激动不已：“谁也没有想到，这种事情竟然会发生。公开密钥加密算法突然就问世了，一点预兆也没有，之前没有任何人能从数学上证明这种算法真的存在。这是一项重大发现——是科学史上的一大奇迹。”

非对称加密算法的问世，也给万维网带来了一场革命。在此之前，尽管银行体系已经建立自己的安全网络，并将其用于通信和金融交易，但相关技术尚未惠及普通大众。1994年，网景

（Netscape）浏览器推出HTTPS——即HTTP的安全版，采用公开密钥加密技术对HTTP数据进行加密。2000年，HTTPS已成为全球网络通行的标准协议。要想判断某个Web网页是否采用了安全的HTTP，只需要查看它在浏览器中显示的URL地址就可以了。凡是URL地址以“https”开头的网页都采用了安全的HTTP。

正因为有了非对称加密算法的保驾护航，电子商务才得以兴起。无论是商品交易、银行操作、竞价拍卖还是证券交易，都可以在网上进

行。众多企业纷纷投入网上运营，甚至开始销售虚拟产品，从中赚取巨额的收入。尼古拉斯·库尔图瓦对非对称加密算法的重要性没有任何怀疑。“整个数字经济都是建立在密码学的基础之上。没有密码学，就没有安全的网络。”

形形色色的网络诈骗

尽管加密算法是非常重要的网络安全工具，但如果你把重要的信息发到了错误的计算机上，就会深陷于麻烦之中。比方说，有的计算机会假冒银行服务器，欺骗用户及其Web浏览器发送银行账号信息。（由于用户加密信息时使用的是对方的公开密钥，因此对方可以将信息解密。）

在计算机安全领域，有一个常见的说法，那就是：在未经身份验证的情况下发送加密信息，就相当于在伸手不见五指的小巷里见陌生人——无论发生什么事情，也不会有任何目击者。作为密码破译者，库尔图瓦对此再清楚不过了：“到目前为止，身份验证比信息加密要重要得多，几乎不可能从中作假。在当今时代，要想将重要的信息完全保密，已经变得难上加难。但是要想进行身份验证，乃至在出错的时候进行补救，都是轻而易举的事情。”

身份验证的方法有很多种，比方说可以采用密码、指纹扫描仪、软件。不过密码、指纹扫描

仪和软件都可以破解。还有一种常见的方法就是，由值得信赖的第三方签署“证书”，验证Web网站的真实性。幸运的是，上面所说的加密算法也可用于身份验证。此外，还有更高级的方法可实现相互身份验证，使客户端和服务端彼此明确对方的身份。

不幸的是，在这场网络安全保卫战中，普通人往往是弱势的一方。犯罪分子具备娴熟的“社会工程学”技巧，行骗手段层出不穷。最常见的手段就是“网络钓鱼”——骗子假冒银行，给你发送邮件。只要你点击其中的链接，页面就会跳转到骗子建立的网站，网站看起来和真的银行平台极其相似，只不过URL地址稍有不同。骗子通过这种方式诱使你输入账号、密码等信息，进而将你账号中的现金全部取走。其他常见的手段还包括电信诈骗、网银升级诈骗等等。比方说骗子会给你发送一条支付确认短信，但其中涉及的交易却并非由你本人发起；再有就是巧立各种名目——让你加入一个不存在的社交网络群组，为非洲的重伤 / 绝症患者提供捐助，升级网银账号、更新网银密码等等，从而达到盗取账号、骗取钱财的目的。甚至连一封看起来非常友善、标题写着“我爱你”的邮件其实也可能潜藏着病毒，只要你运行了带病毒的附件，通讯簿里的所有联系人也会跟着受到牵连。

对此，汉德利表示，这并不是什么新问题。“诈骗犯罪在互联网发明之前就已经很猖獗了，你不能指望它在新的平台上就会有所收敛。”

不过，柯尔斯坦认为，这个问题依然值得重视。“以前我们以为，网络安全仅仅关乎科技，公开密钥加密算法已经很可靠了——剩下的就只是开发应用软件而已。事实上这只是冰山一角，真正需要做的事情还有很多。现在我们意识到，网络安全不仅仅关乎科技，还关乎于人。”

此外，计算机系统遭受攻击而瘫痪的风险越来越大，这样的威胁不仅仅来自黑客，还有可能来自敌对国家。2008年，格鲁吉亚遭到“有组织的网络攻击”，当时正值俄格冲突期间。2010年，伊朗核电站被黑，据说，此次攻击从技术上看，“几乎可以肯定是别国政府机构所为”。2011年，一份报告显示，美国国家航空航天局的安全系统存在漏洞，以至于卫星已完全暴露在黑客的攻击之下。报告称：“……网络服务器将加密密钥、加密密码、用户账户信息暴露给了潜在的攻击者。”

柯尔斯坦发现，自己有时候也卷入了这些事情的风口浪尖，因为他有一个项目隶属于北约和平与安全科学项目（Science for Peace and Security Programme），其任务是在高加索和中亚地区建

立学术互联网（Academic Internet）。“格鲁吉亚当年遭受大规模的网络攻击，所有的Web服务器都瘫痪了。我们有一台项目服务器是最先恢复正常的，后来我们用它来帮助其他服务器重新联网。这件事情让北约意识到，我们这些搞科研的在关键时刻也是能发挥作用的！”

面对新型的电子战，政府将采取什么样的措施，这一点还有待观察。与此同时，我们也需要思考一个问题：为了换取网络安全，我们愿意在多大程度上牺牲隐私？这是一个艰难的权衡。要想提高身份验证方法的安全性，就必须暴露更多的私人信息，让计算机采集指纹、虹膜、甚至DNA。但是私人信息暴露得越多，就越有可能被人滥用。人们愿意让医疗中心采集自己的DNA数据，以便享受更好的医疗服务，但却不愿意让保险公司利用它来检测自己的患病风险，进而将某些疾病从承保范围中移除，或者将保费大幅提升。人们愿意在社交网站上和朋友分享自己的好恶，但却不愿意让营销公司利用这一点来投其所好，推销产品。“人们正在用隐私来换取安全，”汉德利表示，“这是一个艰难的取舍。隐私对于一个人来说事关重大。但人们对隐私的重视还远远不够。”

如今，为数众多的互联网企业正在竭尽全力为用户提供强大的安全保护功能。搜索引擎谷歌

就是其中的一员。谷歌可以抓取大量互联网页面的内容，当它检测到某个网站包含“恶意代码”时，就会警告用户，访问该网站可能会损害计算机。然而，越是依赖这样的外部保护，我们需要牺牲的隐私就越多。毕竟，要想知道访问特定的网站是否会损害计算机，就必须向相关企业透露自己即将访问的网站。如果企业将我们所有的浏览记录都收集起来，就会形成一笔庞大的数据资产，至于这笔资产可能会被用来做什么，着实让人不敢想象，“显然，每一项网络安全技术都有可能被人滥用，造成可怕的后果。”柯尔斯坦表示。

数字生活

互联网和万维网正在从方方面面重塑我们的生活。从20世纪90年代起，很多人发现，开互联网公司似乎有利可图。公司名只要以e开头（比如e.Digital），或者以“com”结尾（比如boo.com），立马就能起到身价倍增的效果。大量的互联网企业如雨后春笋般涌现，它们的销售预测不切实际，股价估值水分过高，由此催生了巨大的泡沫。2000年，互联网泡沫骤然破裂，人们发现，所谓的“.com”企业也不是无坚不摧，许多公司纷纷倒闭。

不过，时至今日，互联网依然引领着商业的

未来。成长为巨头的互联网公司也不在少数。据估计，2010年Facebook的市值在250亿美元左右；2011年初，谷歌的市值达到了令人瞠目结舌的1,900亿美元；更有甚者，苹果的市值在同一时期突破了3,000亿美元大关，其市值的增长主要得益于两个方面，一是新推出的移动计算设备，二是广阔的数字音乐市场和应用市场，这两个方面都与互联网息息相关。

每一款新开发的操作系统上都集成了浏览万维网、访问社交网站的功能，良好的互联网连接环境是所有计算机必不可少的重要性能，不管它们是台式机、笔记本、平板计算机还是手机。各大厂商的游戏机都利用互联网平台运行多人游戏。大多数零售巨头已支持在线购物。

互联网一开始只是众多网络中的一种，如今却已占据统治地位，并以锐不可当之势将其他网络并入其中。马克·汉德利表示：“传统的电话网正在消失，并逐渐成为互联网的一部分。蜂窝网络[\[27\]](#)迟早也会融入互联网。”

2011年，为了顺应新世纪的潮流，英国电信集团（BT）开始建立基于互联网的通信网络，关闭传统的电路交换[\[28\]](#)电话系统。用户的电话服务不会受到影响；座机依然可以接通墙上的插座。但是电话线的另一端与之前截然不同，用户的声音以数据的形式在互联网上传输。“新的通

话系统完全以互联网技术为基础，”汉德利表示，“信号都转换成了数据包，不再在电路中传输了。”当然，在此之前，大多数国际长途电话网都至少铺设了部分互联网线路。

信息的自由流通已经在改变人们的生活面貌。有了互联网，人们可以在线存储信息，随心所欲地安排工作时间和地点，带着导航设备走遍天涯海角，事先观看旅行目的地的照片和视频，随时随地关注朋友的最新动态，通过各种各样的方式沟通交流。如今，娱乐节目已可以按需播放。对此，汉德利跟我们讲了他自己的切身体会：“我儿子今年十岁，他现在不看广播电视，只看iPlayer。他对电视节目的固定播放时间已经没有概念了，因为现在想看什么节目随时都可以看。”

如今，便携式互联网设备的增长似乎已经到了永无止境的地步，由此看来，互联网的未来似乎会朝着无线化的趋势发展。“互联网将呈现出全方位的稳定增长态势。无线网络将给人类生活的方方面面带来巨大的影响。至于我们能否全盘接受，那就是另一回事了。我们需要警惕的是，科学技术是一把双刃剑。”柯尔斯坦表示。

尽管无线网络是互联网未来的发展趋势，但这并不意味着互联网的根本技术概念会发生改变。专家们依然坚信，TCP/IP协议不会被时代淘

汰。“我希望，不管新的设计方案有什么潜在优势，遗留系统⁽²⁹⁾（legacy system）都可以凭借久经考验的强大力量将它打败。新技术的开发必须建立在已有设备的基础之上。”汉德利表示。

但是未来的事情难以预料。“我发现，有些事情我预见得了，但有些事情不行。新事物的出现总是让我措手不及，”汉德利表示，“有一点我比较清楚：我预见不了哪些应用会在市场上走火。每次出现什么新的杀手级应用⁽³⁰⁾，我都觉得很突然。它们总是在出乎意料的地方出现，并不走主流的路子，而是属于剑走偏锋的类型。”

一个简单的计算机网络实验、一些用于共享信息的软件，二者的结合竟然催生出了一个改变世界的新事物，这一点着实非比寻常。我们创造了一个全球规模的计算机社会，在这个社会中，不计其数的计算机日以继夜地高速传递着各种信息。这张令人称奇的网络超越了国家和文化的界限，具备坚实稳定的性能，塑造着当前世界的经济、社会和政治格局。互联网的发明，是人类在20到21世纪取得的最伟大的成就之一。由于这场席卷全球的电子革命，传媒大鳄鲁珀特·默多克（Rupert Murdoch）不得不大规模重建他的传媒帝国。“互联网的诞生，是我一生中经历的最深刻的变革，也是人类几百年来经历的最重大的变革。有一次我偶然听到两个人在谈互联网。其中

一个人说，‘它是《古腾堡圣经》[\(31\)](#)

（Gutenberg Bible）问世以来最重要的发明。’另一个人说，‘不对，它是文字诞生以来最重要的发明。’”

注释

[\(1\)](#) 吼猴（学名：Alouatta）是一属14种物种的统称，这种猴子舌骨发达，可发出巨大吼声，1.5千米以外都可听见。

[\(2\)](#) 模拟通信是一种以模拟信号传输信息的通信方式。非电的信号（如声、光等）输入到变换器（如送话器、光电管），使其输出连续的电信号，使电信号的频率或振幅等随输入的非电信号而变化。普通电话所传输的信号为模拟信号。

[\(3\)](#) 信息论是运用概率论与数理统计的方法研究信息、信息熵、通信系统、数据传输、密码学、数据压缩等问题的应用数学学科。

[\(4\)](#) 调制技术是把基带信号变换成传输信号的技术。

[\(5\)](#) 微波是电磁波谱的一部分，频率范围大约从1吉赫到1太赫。微波是电视、电话以及数据传送的主要载体，资料可以通过微波在地球上的不同站点之间传送，也可以在地球与卫星之间传送。

[\(6\)](#) 1947年，英国将中学从3年制改为4年制，学生因此能够选修更多课程。1951年英国政府决定用GCE取代较早的HSC（高级中学毕业证书，Higher School Certificate）和SC（中学毕业证书，School

Certificate) 系统。在GCE制度中, 高中毕业生(16岁) 参加普通程度考试(GCE 'O' Level), 大学预科毕业生(18岁) 参加高级程度考试(GCE A-Level)。

(7) 见注释115。

(8) 径流是指降雨及冰雪融水在重力作用下沿地表或地下流动的水流。

(9) 欧洲核子研究组织是世界上最大型的粒子物理学实验室, 也是全球资讯网的发祥地, 位于瑞士日内瓦西部接壤法国的边境。

(10) 地震台阵(seismologic array) 是根据研究目的, 在一定研究区, 按某一规则(十字形、圆形、方形等) 布设的一组地震仪。

(11) 协议转换是一个通信系统为向另一系统进行传输而进行的电子信号或数据格式的转化过程。

(12) 网络协议就是为计算机网络中进行数据交换而建立的规则、标准或约定的集合。

(13) 抽象是指, 去除掉被研究对象中与主旨无关的次要部分, 或是暂时不予考虑的部分, 而仅仅抽取出于研究工作有关的实质性内容加以考察。

(14) 土电话的原理。

(15) 本段专有名词的译法及概念讲解参考了第三届社会网暨关系管理研讨会(2007年) 集刊论文《农民工社会网络的连接倾向性》, 杜海峰、任义科等著。此论文为国家自然科学基金(批准号: 70671083)、国家社会科学基金(批准号: 05ARK001)、教育部新世纪优秀人才支持计划(批准号: NCET-04-0931)、西安交通大学“985工程”二期重点项目(批准号: 07200701)、美国Santa Fe研究所国际项目、斯坦福大学CDEHA种子基金(批准号:

AG017253-06) 联合资助项目。

(16) 源节点是指充当信源发送原始数据包的网络节点。

(17) 目的节点也就是充当信宿接受数据包的网络节点。

(18) Marconi professor of communications, 马可尼是荣誉称号, 得名于无线电通信的创始人伽利尔摩·马可尼 (Guglielmo Marchese Marconi)。

(19) 根服务器主要用来管理互联网的主目录, 全世界只有13台。1个为主根服务器, 放置在美国。其余12个均为辅根服务器, 其中9个放置在美国, 欧洲2个, 位于英国和瑞典, 亚洲1个, 位于日本。所有根服务器均由美国政府授权的互联网域名与号码分配机构 ICANN 统一管理, 负责全球互联网域名根服务器、域名体系和IP地址等的管理。

(20) 顶级域 (或顶级域名) 是互联网DNS等级之中的最高级的域, 是域名的最后一个部份, 即是域名最后一点之后的字母, 例如在example.com这个域名中, 顶级域是.com。

(21) 为了保持网络的稳定性, 在多台交换机组成的网络环境中, 通常都使用一些备份连接, 以提高网络的健壮性、稳定性, 这里的备份连接也称为备份链路或者冗余链路。

(22) 瑞典法律规定一年中的几天为升旗日, 瑞典国旗需要悬挂于所有公共旗杆和建筑物上。

(23) 指通过电子联系将相关的信息片断连接起来, 以便用户方便地取用。

(24) 每个应用层协议都是为了解决某一类应用问题, 而问题的解决又往往是通过位于不同主机中的

多个应用进程之间的通信和协同工作来完成的。应用层的具体内容就是规定应用进程在通信时所遵循的协议。

(25) 语义网是对未来网络的一个设想，简单的说，它是一种能理解人类语言的智能网络，不但能够理解人类的语言，而且还可以使人与电脑之间的交流变得像人与人之间交流一样轻松。

(26) 智能卡（Smart Card）：内嵌有微芯片的塑料卡（通常是一张信用卡的大小）的通称。

(27) 蜂窝网络、流动网络或移动网络是一种移动通信硬件架构，分为模拟蜂窝网络和数字蜂窝网络。由于构成网络覆盖的各通信基地台的信号覆盖呈六边形，从而使整个网络像一个蜂窝而得名。

(28) 电路交换，就是在通信时建立（即联接）电路，通信完毕时拆除（即断开）电路。至于在通信过程中双方传送信息的内容，与交换系统无关。

(29) 在计算机更新换代的过程中，怎么处理和利用那些历史遗留下来的老系统，成为影响新系统建设成败和开发效率的关键因素之一。我们称这些老系统为遗留系统。遗留系统的定义有两个：[1]遗留系统是我们不知道如何处理但对我们的组织又是至关重要的系统。[2]遗留系统是指任何基本上不能进行修改和进化以满足新的变化了的业务需求的信息系统。

(30) 杀手级应用的特点是：不仅非常有用，而且让消费者愿意为之专门购买某个硬件或软件产品。

(31) 1456年在梅因斯所印的第一本以活字印刷的拉丁文圣经，据说为古腾堡所印。

计算机把我弄哭了

当工具已经强大到万能的地步，怎样设计才能使它得到最高效的利用？如果没有人机互动，再聪明的机器对于人来说也形同废铁。人类需要创造界面，才能与自己亲手打造的数字大脑进行互动。物理界面能将操作者的动作转换为指令。可视化界面则能通过直观的图形，在人机之间搭建双向交流的桥梁。

一旦界面技术发展成熟，人们或许会沉湎于机器的数字宇宙中，惊艳于不同凡响的全新体验。又或者，计算机会与人的生活浑然一体，就像眼镜一样稀松平常。一旦界面能够表达情感，它们或许会对人的喜怒哀乐做出反应，在人们恐惧不前时给予激励，在人们伤心失意时给予安慰。

然而，人机交互技术究竟应该走多远？人类已经越来越像赛博格（Cyborg）——即人与机器的结合体。如果计算机知道我们所有的秘密，我们怎样才能保护自己不受侵害？

大厅里树着一块黑白显示屏，屏幕上有一张面孔慢慢地消失在背景中。讲台上坐着一名身形瘦削的中年男子，他的头发整齐地梳在脑后，右耳上夹着一款样式现代的带话筒耳机。他操着一口硬铮铮的俄勒冈口音，说起话来带有20世纪60年代的典型特色。这次演示的内容是他在斯坦福大学的研究项目。“大家都是搞科研的人，试想一下，要是给你在办公室里配一台计算机，让它整天保持运转，随时处于待命状态……”说到这

里，他停了一下，面带微笑地看了看台下观众，“换句话说，它可以根据人的指示做出回应。这种技术可以产生多少价值，你们估量得出来吗？”

看得出来，他很激动，在成千上万人面前演讲还是有些紧张。不过多年的授课经验摆在那里，从他的语音语调就能听出，他是个老手。“这么多年来，我们斯坦福研究所的增强人类智能研究中心（the Augmented Human Intellect Research Centre）一直致力于研制这样的机器。”

“幸运的是，我们的成果和技术本身就已经非常引人入胜。我会多做一些演示，少讲一些空话，把项目中的亮点直观地展示给大家。接下来，我给大家演示一下我们的技术成果，”说着，他指了指摄像机，“废话先不多说，我们直接进入正题。”

他的声音从话筒里传出，在整座大厅里回荡。时值1968年12月，秋季联合计算机会议（the Fall Joint Computer Conference）正在旧金山布鲁克斯大厅举行。台上的演讲者正是道格拉斯·恩格尔巴特（Douglas Engelbart），他一次性地演示了数量空前的创新成果，在会场上掀起了一轮高潮。

恩格尔巴特1925年出生于俄勒冈州波特兰市，在三兄弟中排行第二。母亲热爱艺术，父亲

开了家收音机修理店，不过恩格尔巴特并未因此而迷上电子器件。“小时候总觉得电子设备很神秘，看起来怪吓人的。”他后来回忆道。九岁时，他的生活因为父亲的去世被彻底打乱，一家人搬到了威斯康辛州约翰逊·克里克市（Johnson Creek）的乡下。尽管生活上出现了重大变故，恩格尔巴特在学习上依然出类拔萃（只不过他学习的目的并不是为了考高分，而是为了理解事物运作的方式）。他渐渐对雷达技术产生了兴趣。17岁时，恩格尔巴特从波特兰的富兰克林高中

（Franklin High School）毕业，进入俄勒冈州立学院——即后来的俄勒冈州立大学（Oregon State University）攻读电子工程学。他发现自己很难抉择专业方向。“入学才两三个月，我就对学院的教授失望透了，后来我意识到，自己对他们的心里期许本身就不对，没有把他们看成教授，而是在以父亲的标准要求他们。”恩格尔巴特尚未完成学业，就被征召入伍，成为了海军技术兵。新兵训练结束后，他随部队驻扎到了菲律宾。

在前往驻地的路上，恩格尔巴特来到一座名为莱蒂（Laiti）的小岛，歇息了几天。他在岛上发现了一座由竹子搭建的高脚屋，这座高脚屋是红十字会在当地管理的图书馆。恩格尔巴特在馆内浏览图书和杂志时，读到了万尼瓦尔·布什

（Vannevar Bush）在一期杂志上发表的文章——

《正如我们所想》（As We May Think）。布什在文中大胆提出设想，认为机器将来可以帮助人们存储和调取所有科学知识。这篇文章对恩格尔巴特的思想产生了深远的影响。

恩格尔巴特服完兵役后，回到学校继续完成了学业。1948年，他进入美国国家航空航天局的艾姆斯研究中心（Ames Research Center）担任电子工程师，为各种研究项目制造和维护电子设备，平常一有时间就会去斯坦福大学的图书馆充电。此外，他还开始学习跳舞，生性羞怯的性格渐渐有所改观。他遇到了心仪的女孩子，并与之订婚。不过求婚后的第二天，他突然发现自己的生活没有目标，于是决定订立一个人生规划。他的大体目标是赚钱养家，造福社会。但是，怎样做才能造福社会呢？“当时脑子里突然闪过一个念头，觉得世界上有很多问题太复杂，解决它们的办法也太繁杂，”恩格尔巴特回忆道，“很多问题的解决时限越来越短，紧迫性也越来越强。如果把这两个因素综合在一起考虑，我们就会发现，问题的复杂性和紧迫性是考验人类组织机构办事能力的重要指标。如果能开发简单快速的办法处理复杂而紧迫的问题，那就能从根本上为人类做出一些贡献。这个想法一直萦绕在我的脑海里。”

“我的思路很快就展开了。估计当时还不到

一个小时，我就构想出了一台机器，它有一块巨大的CRT显示屏（CRT即阴极射线管），上面显示着五花八门的符号，这些符号可以是人们用过的，也可以是以前从来没有发明过的。计算机可以操纵这些符号，而用户可以通过各种装置操纵计算机。操纵装置的工程结构很简单，可以是杠杆、旋钮、按钮、开关，凡是你能想到的任何工具都行。它们可以传达用户的指令，使计算机做出反应。”

恩格尔巴特不是数学家，他甚至连计算领域的行家也算不上，但他深知，如果计算机可以运算，那么这台巨大的运算机器除了处理庞杂的问题，一定还能做别的事情。恩格尔巴特意识到，如果人人都有一台计算机，再加上显示屏和精巧的输入设备，那就可以实现前所未有的工作方式。他甚至设想了协同工作的模式。“我脑海里浮现出了一些非常清晰的画面，画面中有一群人坐在不同的办公室里，他们属于同一家公司，办公室里配置了相似的工作站，每一个工作站都连接着同样的计算机系统。人们可以通过计算机系统共享信息，紧密协作。我还设想了一种可能性，那就是，一旦计算机被发明出来，很多新的技能和思考方式也会相继问世。”

为了实现自己的新设想，恩格尔巴特决定回到学校继续深造。他选择了入读加州大学伯克利

分校，因为那里的计算机学科建设方兴未艾。恩格尔巴特利用在校时间深入钻研了计算机，并于1956年获得了电子学博士学位。执教一年后，他决定将自己的部分发明加以利用，申请专利，并组建公司开发这些专利。可惜他的公司未能长久经营，于是恩格尔巴特转而加入了斯坦福研究所。入职后，他取得了更多的发明专利，工作进展较为平顺，不过，他依然想实现当初的设想，只可惜没什么人支持。1962年，恩格尔巴特将自己的想法写成了正式的文稿，题为《增强人类智能：一个概念框架》（Augmenting Human Intellect: A Conceptual Framework）。当时，他在美国航空航天局有一位朋友，名叫罗伯特·泰勒（Robert Taylor）。在泰勒的帮助下，他终于拿到了科研经费，由此开展了自己的项目。

恩格尔巴特的科研经费来源于美国高级研究计划局，该局还派来了两名研究人员前来帮忙，他们分别是利克莱德和伊凡·苏泽兰（Ivan Sutherland）。苏泽兰也是人机互动领域的先驱，他在麻省理工学院攻读博士期间，就已经发明了一款功能强大的绘图软件，名为“几何画板”（SketchPad）。只要使用光笔和一组按键，就可以在“几何画板”上绘制直线和曲线，程序会对绘制出来的线条进行自动调整。用户可以将作好的设计图放大，图稿放大后，超出屏幕的部分

会被自动“切掉”。此外，用户不仅可以旋转、复制和移动图片，还可以拖拽图形上的线条和点，进而改变图片的局部尺寸。这些实用的功能到如今已成为所有绘图软件包和计算机辅助设计系统的标准配置。有了如此出色的研究履历，苏泽兰自然是恩格尔巴特当之无愧的左膀右臂。

很快，恩格尔巴特创立了增智研究中心（Augmentation Research Center，简称ARC），开始招贤纳士。接下来几年里，他发明了计算机鼠标，并将它与已有的设备进行比对测试，还提出了一系列领先于时代的惊人设想。1968年12月，在秋季联合计算机会议上，恩格尔巴特及其研究小组在合作伙伴的协助下展示了他们的研究成果。此次演示因其生动、直观的形式和前瞻性的成果而被人冠以“演示之母”（The Mother of All Demos）的美名——这样的赞誉可谓实至名归。

图形用户界面的诞生

时光倒退回那历史性的时刻。恩格尔巴特坐在旧金山布鲁克斯大厅的讲台上，面前摆着一个形似大托盘的容器，托盘中间摆着一个计算机键盘。右边有样东西看着明显像是鼠标，只不过更笨重一些。鼠标控制着投影屏幕上的指针，这个指针想必当今时代的人都很眼熟，恩格尔巴特喜欢称它为“虫子”。托盘左侧放着一排“和弦电

键”，看起来像五个钢琴键，可以以不同的排列组合按键打字。换言之，同样是打一个字符，正常的键盘只需要按一个键，“和弦电键”则需要按多个键。这种操作方法可能学起来比较困难，所以“和弦电键”没能经受住时间的考验。

在接下来一个小时的时间里，恩格尔巴特首先演示了文字处理器的操作方法，比如用计算机键盘敲击打字，用鼠标复制粘贴文本，标题自动编号，文字替换，隐藏文档部分内容，项目分类，排序，选择性地显示列表中的项目。此外，演示内容还包括简单的交叉引用（超链接）——点击文字，即可跳转到相应的关联信息；点击不同的链接，即可跳转到不同的页面，或打开新的图片。恩格尔展示了一张带动态链接的简单图片，点击图上的某个单词，即可打开一个新的文本页面。

演示完文字处理器后，他连线数英里外的同事，举行了视频会议。他创造性地采用阴极射线将视频画面投射到投影幕上，用一个窗口显示。在画面中，他的同事们在展示编程语言和编程环境之余，还进行了一系列演示，比如按照打印格式排版文档，在协作者之间共享文件、乃至显示屏，用鼠标在屏幕上画图。当时没有人见过这么新奇的玩意。在1968年，观看这场演示就像见证一个又一个奇迹。

演示的最后，恩格尔巴特提到了一个新项目——美国国防部高级研究计划局建立的一个试验网络。“在这个网络中，我们将尝试开发一项特别服务，给人们提供相关的网络信息，以便运营这类网络。我们计划提供的信息包括：服务内容有哪些，面向哪些人群，采用什么样的协议才可以登录，当天登录的用户有哪些，哪里可以找到用户指南和系统描述文件。”

恩格尔巴特的确在阿帕网上提供了简单的信息服务，而且在互联网的早期发展阶段发挥了重大的影响。如果事情按照这样的趋势发展下去，他兴许就能发明万维网，或者与之非常类似的网络了。然而不幸的是，在接下来的十年里，情势急转直下。随着管理层的洗牌和赞助问题的涌现，斯坦福研究院的内部乱成一团，四分五裂。1977年，恩格尔巴特遭到解雇。

被解雇后，他试图在一家名叫蒂姆沙尔（Tymshare）的新公司继续开展工作，新公司致力于将增智研究中心（ARC）研发的产品商业化。但是很快，恩格尔巴特发现自己遭到排挤，无法发挥创新才能。虽然这对他来说时运不济，但对另一个研究中心来说，无疑是天赐良机。原来恩格尔巴特手下的很多ARC小组成员后来都进了施乐（Xerox）公司的帕罗奥多研究中心（Palo Alto Research Center，简称PARC），很快，该中

心就成了业内研究领域新的领头羊。

早在1969年，帕罗奥多研究中心的员工就已经发明了激光打印机。有了ARC小组精英团队的加持，加上新经理鲍勃·泰勒的英明领导，计算机界面的电子开发很快就开始突飞猛进。“窗口”的概念进一步完善，一个程序可以有多个窗口，每个文档占用一个窗口，后来又发展到独立程序可以各自拥有专属的窗口。菜单（只有在点击关键词的情况下才会出现的下拉列表选项）和图标

（可以代表文件、文件夹、程序或“最小化”的窗口的小图形）相继问世。单选按钮（选项列表中只有一项可选）和复选框（选项可以勾选或取消勾选）使用户与计算机的互动更加直观。所有这些创意的实现，都离不开一个指针设备——鼠标。新风格的界面称为WIMP（窗口、图标、菜单、指针设备的英文缩写）。它与传统的“命令行”界面天差地别，在传统界面下，用户只能通过晦涩难懂的命令代码与计算机互动。

不久后，计算机操作系统开始采用基于上述概念的图形用户界面（GUI）。1981年，施乐推出了Star工作站（施乐8010信息系统），这是第一款采用图形用户界面的商用机。它探索了WYSIWYG（所见即所得）的设计理念。计算机上的文档打印出来是什么效果，在显示屏上就是什么效果，这使得用户开展工作的方式变得更加

简单、直观。Star工作站还具备网络、电子邮件功能，支持不同文档内容的插入（比方说，用户可以将表格软件中的图表插入到文字处理器的文档中）。

苹果公司的创始人史蒂夫·乔布斯对Star工作站的图形用户界面极感兴趣。他领导项目团队，开发了一款新的个人计算机，称为苹果Lisa。苹果Lisa于1983年正式发布。没过多久，另一款计算机麦金塔（Macintosh）也于1984年亮相市场。苹果的创意引发了各大高校和计算机公司的纷纷效仿。1984年，麻省理工学院为UNIX系统开发了X-Windows图形用户界面。1985年，雅达利ST（Atari ST）和Amiga计算机分别上市。在那个年代，IBM个人计算机虽然是办公设备的标配，但是复杂的MS-DOS命令行界面已使它渐渐落伍于时代。1985年，一款适用于MS-DOS系统的图形用户界面横空出世，这就是日后名声大噪的Windows。经过反复改进，1990发布的Windows 3.0和1992年发布的Windows 3.1成为了风靡全球的个人计算机系统。

计算机行业的面貌从此彻底改变。随着个人计算机公司之间的竞争日益加剧，最好的设计理念往往能够在模仿和创新中保留下来，并不断改进。如今，我们使用的操作系统大多是基于Windows平台。虽然底层架构的复杂函数依旧可

以通过命令行界面执行，但是简洁美观的图形用户界面是未来发展的大势所趋。随着手机和平板计算机的日趋流行，移动计算也益发重要，后WIMP时代的计算机界面不断涌现。苹果率先在PowerBook笔记本上采用了触控板技术，同时为为数众多的新产品中不断创新，始终走在时代的尖端前沿。譬如，该公司采用了多点触控显示屏，可实现滑动翻页、双指缩放⁽¹⁾、双指旋转⁽²⁾、双指滚动⁽³⁾等多种触控体验，让触屏机的操纵得心应手。

单从行业影响力来看，恩格尔巴特已然过气。但直到2010年，他依然活跃在人们的视野里。他创立了道格·恩格尔巴特研究所（Doug Engelbart Institute），但颇具讽刺意味的是，该机构如今已是斯坦福研究所的一部分。它的宗旨是“增强人类处理复杂、紧迫问题的能力”。恩格尔巴特的眼光依然和几十年前一样犀利。他表示：“人可以表达自己的思想，但真要比较起来，我觉得计算机的灵活性要比人强得多。我们现在碰到的问题越来越多，也越来越复杂，如果我们能够切实地把这些复杂的情况理清楚，那将是非常重要的成就。人们总是说，我们已经做得很不错了，可我还是要说，我们还有很长的路要走。”

用第二双眼看世界

20世纪60年代对于计算机科学家来说，是一段忙碌的时期。从某些方面来看，恩格尔巴特一直比较幸运，高级研究计划局信息处理技术科

（Information Processing Techniques Office，简称IPTO）的连续三任领导都和他关系很好。第一任领导利克莱德是个独具慧眼的人，他不仅在网络领域有很深的造诣，而且在人机交互的问题上也颇有见地。这种广博的视角使他更容易接受恩格尔巴特的观点和目标，所以批准经费也是理所当然的事情。1964年，苏泽兰接替了利克莱德，他是交互式图像学领域的先驱。1966年，苏泽兰的继任者泰勒对网络和图像学都很有兴趣，他曾经参与过阿帕网的研制项目，后来又担任了帕罗奥多研究中心计算机科学实验室的主任。

在这些先驱当中，贡献最突出的要属苏泽兰。苏泽兰曾经在高级研究计划局工作过一段时间，但他真正感兴趣的领域其实是计算机图形学^{[\(4\)](#)}。早在学生时代，他就已经发明了绘图软件“几何画板”，而今，他着眼的是更加宏伟的目标。

伊凡·苏泽兰1934年出生于美国内布拉斯加州黑斯廷斯市（Hastings）。他和哥哥伯特（Bert）从很小的时候起，就对计算机产生了浓厚的兴趣，平时喜欢用继电器自制机器。他们的父亲是

一位土木工程师，他给兄弟两人的成长带来了很大的影响。伊凡长大后，一家人搬到了纽约郊区的斯卡斯代尔（Scarsdale），兄弟两人开始为计算机先驱、美国计算机协会创始人埃德蒙·伯克利（Edmund Berkeley）打零工。虽然只能拿到最低水平的工资，但兄弟俩做得不亦乐乎。伊凡给伯克利的个人计算机编写了第一款除法程序，称为“西蒙”（Simon）。“我写的程序可以做一部分除法运算，被除数最大可以到15，除数最大可以到3，”苏泽兰说，“在我写程序的过程中，伯克利给了提供了许多指导，讲解了许多计算方面的知识。”

在此期间，伯克利把苏泽兰兄弟介绍给了还在贝尔实验室工作的克劳德·香农。“我和伯特去拜访了香农，那一天我们相处得非常愉快。”后来，这层关系也给他们带来了很多便利。兄弟二人受到香农的启发，为伯克利发明了一只会走迷宫的机器老鼠。伊凡中学毕业后，考取了卡耐基理工学院（Carnegie Institute of Technology）——即现在的卡耐基梅隆大学（Carnegie Mellon University），攻读电子工程专业。随后，他又在加州理工学院（California Institute of Technology）获得了电子工程专业的硕士学位。出色的教育背景使他获得了在IBM公司打暑期工的机会。然而，随着苏泽兰对计算机的研究日益

深入，他开始意识到，麻省理工学院的计算机系才是数一数二的顶尖计算机院系。于是他联系了当时已在该校任教的香农。“香农还记得我们兄弟俩，他把我邀请到他家里去了，”苏泽兰回忆道，“如果不是因为他的提携，我就无法取得今天的成就了。”伊凡的博士研究课题是开发绘图软件“几何画板”，如上文所说，它是个令人称奇的交互式设计程序。当时担任答辩委员会主席的正是香农，此外，人工智能领域的先驱马文·明斯基也是答辩委员会的成员。

伊凡完成博士课题后，加入了预备役军官训练营（Reserve Officer's Training Corps，简称 ROTC），当时美国很多重点高校的学生都必须服兵役。伊凡在陆军部队为一名上校当技术兵，同时为美国国家安全局（National Security Agency，简称NSA）制造计算机显示器。服役一年后，他被授以中尉军衔，并调往国防部高级研究计划局，担任信息处理技术科的负责人。他在这个岗位上干了两年。在此期间，他利用职务之便积累了宝贵的人脉，结交了大量的计算机科学家，其中包括专攻图形学的戴维·埃文斯（David Evans）。但是苏泽兰并不想在信息处理技术科待下去。“我有自己的职业规划，当初入职的时候就说过，我只会在这里干两年。既然两年期已满，那应该做点别的事情了。”苏泽兰把信息处

理技术科的工作交接给鲍勃·泰勒后，自己接受了哈佛大学的终身教职。他终于可以按照自己的想法开展尖端研究了。

当时，新开发的阴极射线管显示器上已经可以实现二维成像，只不过，成像的范围仅限于线框图形（物体的轮廓都是细细的直线）。但是既然二维图像都可以显示，为什么三维图像就不能呢？为什么不利用人的两眼视角差，来制造图像的空间立体感呢？实现三维成像后，为什么不更进一步，让三维图像随着用户视角的改变而发生动态的改变呢？“一个人在抬头、低头或者做其他各种头部动作时，他所看到的真实物体的图像就会发生改变。显示器中的三维图像也应该具备这样的效果。”苏泽兰表示。

这就相当于让用户体验虚拟现实（virtual reality）。用户只能看到由计算机创造的虚拟世界。如果他们在现实世界中来回走动，在虚拟世界中也会这样。

这是一个非比寻常的创意，苏泽兰及其研究小组耗时数年，终于建立起了一个可以实现虚拟现实的工作系统。不过在1968年，也就是恩格尔巴特在计算机会议上做演示的那一年，苏泽兰发布了一篇文章，详细描述了他的头戴式三维显示器。虽然这款显示器的确是头戴式的，但它实在太重，使用时必须用缆索将它吊在天花板的支架

上。此外，它的显像质量非常粗糙，能够显示的最复杂的图形也不过是一座“空房间”，上面有一扇门和三扇窗户——其实，所谓的“空房间”也不过是个立方体而已，上面开了一个长方形和三个正方形。更糟糕的是，显示器内置的超声波位置传感器在使用过程中容易产生累积误差，导致系统有时候必须停机几分钟之后才能正常使用。不过，苏泽兰的头戴式设备虽然存在种种瑕疵，但使用起来还是没有问题的。“用户很快就会习惯戴着显示器看东西。要想环视四周，只要转一下头就可以了。”苏泽兰表示。

尽管显示器的大部分研制工作是在哈佛大学完成，但苏泽兰还是在1968年转职到了犹他州立大学（University of Utah），因为他的朋友戴维·埃文斯在那里创立了计算机科学系。很快，苏泽兰对计算机图形学（乃至整个计算机科学）的影响力开始与日俱增。他的学生丹尼·科恩（Danny Cohen）发明了一款适用于通用计算机平台的飞行模拟器，用于训练飞行员。师生二人共同提出了一种算法，称为科恩-苏泽兰三维直线段裁剪算法（Cohen-Sutherland three-dimensional line-clipping algorithm），这种方法可用于自动“裁剪”图片，使之成适合屏幕大小。后来，埃文斯与苏泽兰携手创立了埃文斯-苏泽兰公司（Evans and Sutherland），该公司在实时图形、3D图形及

打印机语言等领域做了大量重要的工作。不少雇员后来自立门户，创立了非常成功的公司，比方说吉姆·克拉克（Jim Clark）创立了硅谷图形（Silicon Graphics），约翰·沃诺克（John Warnock）创立了Adobe。

苏泽兰有很多学生在校期间便已享有盛名。亨利·高洛德（Henri Gouraud）发明了一种效果逼真的光影渲染技术，称为高洛德着色⁽⁵⁾

（Gouraud shading）。弗兰克·克罗（Frank Crow）发明了抗锯齿技术（anti-aliasing）。由于在3D图像中，受分辨率的制约，物体边缘总是会或多或少地呈现出三角形的锯齿，而抗锯齿就是通过像素混合，对物体边缘进行柔化处理的方法（如今，这项技术在每台计算机和电视上都得到了广泛的应用）。艾德文·卡特姆（Edwin Catmull）是皮克斯（Pixar）的联合创始人，现任迪士尼动画工作室和皮克斯动画工作室总裁。

苏泽兰停止了在计算机图像学领域的研究工作。1976年，他转职到加州理工学院，在那里创立了电子计算学系，当上了系主任。在他的推动下，集成电路设计成为了计算机科学的分支领域。1980年，苏泽兰成立了咨询公司Sutherland, Sproull and Associates。该公司后来被升阳（Sun Microsystems，又译为太阳微系统）收购，成立了升阳实验室（Sun Labs）。甲骨文收购升阳

后，该实验室转而成为甲骨文实验室（Oracle Labs）。苏泽兰曾在升阳担任研究员和副总裁。2008年，他转职到波特兰州立大学（Portland State University），从事异步系统^[6]的研究工作。

相片与网格

计算机图形学在苏泽兰等先驱的推动下迅速发展成熟。随着市场对计算机游戏的需求越来越大，软件算法和计算机硬件的功能也日益强大。正如我们在第010章所看到的那样，如今世界上最快的超级计算机都是基于图像处理器而研制出来的。随着显像技术的日新月异，3D电视和计算机显示屏正在主流市场上大行其道。

尽管技术的发展态势喜人，但总体来说，整个计算机图形学体系还只是基于两个简单的概念：位图和矢量图^[7]。位图又叫做栅格图，简单地说，位图就是数码相机拍出来的相片。它由像素组成，每个像素都被分配一个特定位置和颜色值。位图放大后会变得模糊不清，看起来像是有许多小方格。这是因为，图片放大的其实只是像素点。相机上的光学放大之所以数码放大效果更好，原因在于，数码放大只是将像素点的尺寸放大，并未增加任何细节，而光学放大通过光学镜头结构的变化来实现变焦，就像显微镜一样，

能够将更多的细节展示出来。

与位图不同，矢量图是计算机严格按照运算结果实时绘制出来的。这就好比我们小时候在数学课上做过的事情：在坐标纸上描点，用尺子画线，将点与点连接起来。在矢量图中，所有的图形都是由点和线构成，就连图中嵌入的文字也是如此。它的优势在于，每次放大或缩小图像时，计算机都会重新计算所有线段的长度和位置，然后在维持原有清晰度的情况下，将图像重新绘制出来。也就是说，矢量图不会失真。（伊凡·苏泽兰的几何画板可能是世界上第一款矢量图形设计软件。）

我们可以利用矢量图玩出不少新花样，比如构建复杂的三维模型：可以用三角形、四边形或者五边形组成数以千计的细小网格，让计算机在网格中填充颜色，然后用着色器（譬如高洛德着色算法）渲染出逼真的光影效果。有了大量细小的网格，再加上一些化腐朽为神奇的立体渲染技巧，我们就能将曲面造型构建出来。除此之外，一些简单的构图规律也可以用来制造三维立体感，比方说透视规律：可以绘制透视图，使画面中的物体呈现出近大远小、近长远短、近清晰远模糊的变化特点，同时采用一些巧妙的消隐算法，将当前观察点看不见的三维模型表面隐藏，从而逼真地反映物体的空间形象。计算机图形学

领域有大量令人称奇的美图方法，可以制造精巧繁复而又美不胜收的光照、雾化和渲染效果，模拟水波、皮毛、织物等丰富多彩的质感纹理。如果时间充裕，我们甚至还可以采用光线跟踪技术。为了便于大家理解光线跟踪的概念，我们首先假设屏幕内的世界是真实的，显示器只是一个完全透明的框架，那么屏幕内的世界里应该有哪些光线会透过屏幕射入人的眼睛呢？光线跟踪技术所追踪的，正是那些能够射入人眼的光线。它的原理是从视点出发，逆向追踪光线的传播路径。比如光线会不会从水面反射出去？会不会穿过玻璃，发生折射？物体的瞬间移动会不会造成画面模糊？如今，《阿凡达》（Avatar）、《怪物史莱克》（Shrek）等好莱坞大片大量采用了光线跟踪等计算机图形学技术，用于后期制作。

光线跟踪技术虽然效果逼真，但运算量庞大，实时处理速度慢。对于游戏等实时性要求高的大型复杂场景或虚拟现实系统来说，要想做到实时性与真实感并重，就必须将位图和矢量图巧妙地结合起来。要想用矢量图作出复杂多变、自然度高的图像难度极大，矢量图的线条往往太过干净、平顺，一看就知道是假的。但是位图就不一样了，要想拍出还原度高的数码相片（或者用软件绘制出这样的图像）并不困难。只要像素足够高，位图就可以逼真地展现世间万象的纷繁细

节，比如木材表面的纹理、皮肤上的毛孔、高低起伏的地貌等等。美中不足的是，位图是二维平面图，整个图像由无数个方块构成，而人们喜欢的往往是三维立体图（或曲面）。为了解决这个问题，一个两全其美的办法就是，用矢量图网格图形进行三维建模，然后将精美的位图图案“粘贴”上去（这就是上文提到过的纹理贴图），最后用各种巧妙的渲染手法加以美化。

这样做出来的效果足以以假乱真，人眼已经不可能分辨出它到底是拍摄出来的真实场景，还是用计算机生成的虚拟图像。现在的图形处理器不仅价格便宜，而且功能强大，连普通的手机都可以做出如照片般真实的图像。我们只是受制于自己的想象力罢了。计算机不受现实视角的局限，因此可以按照我们选择的任何风格渲染图像。无论我们想让一部电影或游戏看起来像手绘而成的漫画书，还是莫奈的感人画作，甚至是一只大黄蜂眼里的影像，计算机都可以做到。

计算机上可以显示的图像不受现实事物的局限。它可以展示早已灭绝的恐龙在远古森林里漫步的情形，重现昔日繁华的都市未曾被胶片记录下来的荣光，展现宇宙中最微小的粒子乃至最宏大的星系，甚至还可以窥探人体的组织器官，分析大脑的神经机理。

亦真亦幻的世界

如今，计算机图形学或许已经十分发达，但是虚拟现实技术依然离我们的日常生活非常遥远。自苏泽兰在1968年取得开创性的成果后，头戴式显示器的研究工作一直停滞不前。直到20世纪80年代中期，情况才有了转机。那时候，计算机科学领域出现了一位奇才——杰伦·拉尼尔

（Jaron Lanier），他性情古怪，特立独行，既是科学家，又是艺术家。拉尼尔与人联合创立了VPL研究公司（VPI Research, Inc.），这是世界上第一家销售头戴式显示器和“数据手套”的公司。所谓数据手套，就是一种虚拟现实硬件，用户可在虚拟场景中进行物体的抓取、移动、旋转等动作。“虚拟现实”这个术语就是拉尼尔发明的。时至今日，拉尼尔的影响力依然不容小觑，他给林登实验室（Linden Lab）担任过网游“第二人生”（Second Life）的咨询顾问，还为微软设计了XBOX360的体感周边外设Kinect。

头戴式显示技术还在不断进步，但缺陷依然很多，要想在日常生活中广泛使用尚需时日。目前最好的头戴式显示器不仅价格昂贵，而且还比较笨重。为了克服这些问题，很多虚拟现实研究人员采用了洞穴状自动虚拟系统（Cave Automatic Virtual Environment，简称CAVE）。它可以使用户完全沉浸在计算机创造的虚拟世界中，仿佛身

处一个特制的房间，房间的墙上、地板上和天花板上都显示着三维投影（要看到这些投影，通常还需要借助特制眼镜）。计算机会根据用户的动作改变投影：用户站在崖边探身就可以看到下面的情况，跳起身来就可以看得更远，走路（或乘坐模拟器“飞行”）时就会看到两边的虚拟景物在移动。虚拟现实技术制造出来的幻境非常逼真，用户很快就会有身临其境的感受。他们在现实世界中或许正置身于围墙之内，但却对此浑然不知。在虚拟现实，他们唯一能感觉到的只有眼前的虚拟世界。

梅尔·斯雷特在巴塞罗那大学（University of Barcelona）和伦敦大学学院担任教授，研究方向是虚拟现实与虚拟环境。他领导的实验神经科学与技术虚拟环境实验室（Experimental Virtual Environments for Neuroscience and Technology Lab，以下简称EVENT实验室）开展了一系列令人惊叹的研究工作，对虚拟现实技术进行应用。斯雷特认为，虚拟现实的计算机界面之所以颇具吸引力，其原因主要有两点。“首先，从某种程度上讲，人类在虚拟现实中实现了一个由来已久的梦想，”他表示，“人们总是梦想着脱离平凡的现实生活，去一个能体验各种刺激的地方。虽然虚拟现实技术不可以真的实现这种梦想，但它能从技术上营造一个可以让人随心所欲的幻境，让

人完全沉浸在其中。”

不过，虽然斯雷特之所以研究虚拟现实，也有可能是因为这类天马行空的梦想，但他的研究工作更加注重的是这项技术给人类带来的实际益处。“虚拟现实的另一大魅力在于，它与传统的互动界面给人带来的体验完全不同。同样是与虚拟人互动。如果你是对着传统计算机的显示屏，那么虚拟人看起来就只有人的手指大小，你可能会觉得不真实。而在虚拟现实中，这个人和你差不多高，他就站在你面前，直视着你的双眼，就会有强烈的冲击力，感觉他可能随时都会上来打你，或者亲你。”

只要对完全沉浸在虚拟环境中的用户进行研究，就会发现一个有意思的结果：虚拟现实的投影无需太过真实，就能让用户的行为与在现实世界中毫无二致。任何让用户以为站在高楼上的简单错觉都足以使容易眩晕的人极度紧张。在虚拟现实的演讲场合，即便台下的观众都只是卡通式的虚拟人物，那些害怕在公众场合发表演讲的人还是会发抖。斯雷特的小组在心理学家和神经科学家的帮助下，对这些反应进行了研究。“要想在虚拟环境中制造出站在楼顶向下看的感觉，其实很简单，”斯雷特说。“有恐高症的人在虚拟现实中也可能会发作，会产生非常糟糕的焦虑症状。这意味着心理学家可以在实验环境中对恐高症进行

研究。这可比花上几个月的时间，劝说患者走上楼顶要快多了。”

除此之外，EVENT实验室的研究人员还观察了人在虚拟环境中的其他行为。“最近，我和我指导的博士生潘雪妮（Sylvia Xueni Pan，音译）一起观察了害羞的男性在虚拟世界中和异性互动的过程，”斯雷特笑着说。研究人员出乎意料地发现，当虚拟的异性靠近时，受试者还是会躲开。“这个实验非常成功，”斯雷特解释道。“我们的目的就是证明它对治疗是很有用的——受试者在虚拟现实中的表现和在现实生活中相似。一开始他们有些惊慌，但过一阵子就好了。”

研究人员甚至还探讨了如何利用虚拟现实技术帮助偏执的人克服偏执倾向。之所以会探讨这个问题，是因为研究人员发现，几乎所有受试者都会揣测他们遇到的虚拟人物的想法，而这种揣测往往只是胡思乱想。“他们从虚拟环境中走出来以后，就会讲一些不着边际的想法。曾经有个受试者跟我说，‘我在里头碰到一个虚拟人，每次跟她说话，她都不看我。她肯定看我不顺眼。’其实这个虚拟人的反应纯属巧合，根本不是针对受试者的。”为了探讨这一现象，EVENT实验室的研究人员在一位心理学家的建议下找到了丹尼尔·弗里曼（Daniel Freeman）教授，弗里曼教授是伦敦大学国王学院精病学研究所（Institute of

Psychiatry) 的专家，研究领域是偏执症。在他的帮助下，EVENT实验室的研究人员开展了初步研究，其目的是为了偏执的人（非患者）在虚拟现实中的偏执倾向是否与在日常生活中相同。“答案是肯定的，”斯雷特表示，“这很神奇。我们专门设计了一个虚拟环境，里面的情况我们一清二楚。受试者所处的场景是在一个图书馆里。图书馆里很安静，换做是在现实生活中，人在这种环境下的偏执症倾向会上升。试验结束后，有偏执倾向的受试者就会开始说一些他们胡思乱想的事情。比如‘不管我什么时候扭过身去，他都会开始在背后议论我。我老是听到他在背后小声说话。’‘我在里头碰到的那个商人不喜欢我。’总之就是这类不着边际的想法。”

“借助虚拟现实技术，我们为很多人创造了类似的场景，在这些场景中，虚拟人物对用户的态度完全是中立的，”弗里曼说，“试验结果非常令人满意。受试者的反应各有不同，有一部分是以积极或中性的视角看待他们与虚拟人的互动过程，还有一部分是以偏执的视角揣测虚拟人的行为。”研究小组在进一步的研究中采用了更为逼真的场景。他们预计，类似的研究会给心理疾病的治疗带来突破性的进展。现在已经有公司在为恐飞症和恐高症患者提供虚拟现实疗法了。

虚拟现实技术还有许多重要的应用领域。考

古学家可用它来复原古迹，让参观者亲眼领略昔日恢弘建筑的荣光。民航和军事飞行员经常在飞行模拟器上展开训练。越来越多的医学机构开始运用虚拟现实技术培养外科手术师，毕竟，在虚拟手术台上出错总比在实战中出错要好。远程手术也用到了这项技术——手术师遥控操作机器人开展手术，同时借助虚拟现实设备获得现场的反馈信息，感觉就好像在现场一样。

与此相关的增强现实（augmented reality）技术也流行开来，所谓的增强现实，就是将虚拟的人或物品叠加到用户看到的真实影像上。这项技术实现起来要比虚拟现实容易得多。人们常用的平板计算机、智能手机等移动设备只要内置了小型加速计和摄像头，就可以转换为一个增强现实的“窗口”。比方说，参观博物馆时，只要将展品放在移动设备的摄像头前扫描，就可以看到展品表面叠加的细节数据，说不定还能看到展品在原本的环境中是什么样子。外出漫步时，用移动设备扫描地面，增强现实窗口就可以利用地质学和古生物学数据，将脚下的一层层岩石和化石显示出来。

或许，虚拟现实技术最令人称奇的应用领域莫过于制造“换体”错觉。“虚拟现实可以完成一项非常特别的事情，那就是给人制造新的身体，”斯雷特表示，“原理非常简单。戴上头戴式

显示器以后，你所看到的就是虚拟世界。你可以四处张望，这样就可以看到虚拟世界中更多的地方。很显然，你也可以低下头来看自己。我们通过编程，在真实身体的位置制造了一个虚拟身体，这样你所看到的就不是自己真实的身体了。”

研究人员给用户做出虚拟身体后，用户就会产生强烈的错觉，误以为这就是他们真实的身体。“产生这样的错觉是有原因的，”斯雷特说。“毕竟从小到大，无论什么时候低头看，你看到的都是自己的身体。所以从某种意义上来说，大脑已经形成了惯性——只要低头看，我就能看见自己的身体，这是我的身体——它没有别的选择。所以虚拟现实技术可以导致非常强烈的错觉。”

在三维运动捕捉传感器⁽⁸⁾的推波助澜下，这种错觉意识会被强化。当你运动自己的身体时，虚拟身体也会跟着运动。触觉还能进一步增强这种错觉。如果你看到一个虚拟球体碰到了自己的肚皮，同时又真的体验到了球体的触感，那么你的大脑就会误认为这个球体是真实的，即便它看起来可能不太真实。要想了解人类如何认识自身及世界的问题，虚拟现实将是一个强有力的工具。正因为如此，这项技术已在认知神经科学领域得到日益广泛的应用。

虚拟现实技术还有一些有趣的用途正待开发。试想，假如你代替别人参加某个活动，这种角色转换会给你的态度造成什么影响？“假如有人在现实生活中个子很矮，进入虚拟现实以后，突然发现自己比其他人都要高。这对他待人接物的态度会产生什么影响？假如你对某个群体有偏见，进入虚拟现实后，突然发现自己竟然是该群体中的一员。这将对你的态度和行为造成怎样的影响？这些都是值得探讨的问题。”斯雷特表示。

研究人员还在探索这些问题的答案。无论结果如何，虚拟现实的重要性都是毋庸置疑的。它不仅能够帮助人们认识自我，还比人们以往采取的研究方法要安全得多。很多人已经习惯了沉浸式的计算机游戏，这些游戏都采用了3D显示设备和肢体动作感应技术。也许将来有一天，人们也会习惯于虚拟训练和治疗，以保持思维的活跃和健康。

虚拟现实领域的计算机科学家对于这项技术的未来有他们自己独到的认识。“你要是去请教业内人士对这个行业的未来有什么看法，一般都会得到相同的答案，”梅尔·斯雷特教授微笑着说，“其实答案在电影里都已经放出来了。《星际迷航》里的那种‘全息甲板’就是我们想要的效果。虚拟现实所追求的境界就是以假乱真，人们

知道眼前的景象是幻景，但它不是实景，胜似实景。”

重要的不是做什么，而是怎么做

无论是亦真亦幻的虚拟现实，还是随处可见的图形用户界面，这些交互式计算机界面既是科技发展史上的奇迹，也是计算机科学家的梦想在现实世界中的投射。正因为有了它们，人们才得以通过多种多样的方式得心应手地操控计算机，驰骋天马行空的想象力，在缤纷繁复的虚拟世界中随心所欲地体验极致的享受。只不过，技术给人带来的，并非总是一成不变的美好。

事实表明，人机交互技术也是一把双刃剑。有的时候，计算机外设要是使用不当，也有可能给身体造成损伤。“我们把鼠标之父恩格尔巴特请来演讲时，曾经开玩笑说，那些‘鼠标手’患者应该在讲堂外举横幅抗议。”研究人本技术的安吉拉·萨斯教授说，“长期使用鼠标比长期使用键盘更容易造成手部损伤，而且几率要大很多。”

除了鼠标和键盘之外，还有不少设备被指存在健康隐患，任天堂Wii也未能幸免。任天堂新出的Wii游戏机在市场上广受热捧，它创新性地搭载了无线输入控制设备。但这款无线输入控制设备在给玩家带来全新体验的同时，也有可能造

成关节疼痛，人们把这种病症戏称为“Wii炎”（Wii-itis）。有些玩家在Wii上打网球时，喜欢投机取巧，用手腕的翻转动作代替手臂的大幅挥拍，这样连续打上几个小时后，可能会因为肌腱拉伤而入院。

事实上，每次一有新的输入设备出现，人们的健康就有可能受到冲击。设备的形状、重量、材质乃至操纵模式都有可能是其中的罪魁祸首。Xbox Kinect等体感外设虽然能够让玩家摆脱传统手持控制器的束缚，但这并不意味着它们不会影响玩家的操纵动作。“问题并不在于技术本身，而在于，技术设备会把不自然的操纵动作强加到玩家的身上。”情感计算^[9]专家娜迪亚·贝尔托表示。

由于人与计算机技术的互动问题错综复杂，人机交互（HCI）作为一个全新的计算机科学领域应运而生，并在过去30年里发展成熟。人机交互领域的科学家分为三类：第一类是认知科学家，他们着眼于人的思考方式及人与技术的互动过程；第二类科学家着眼于人的因素，他们研究的问题是，人类为了提高技术设备的性能、安全性和用户体验而与周边环境采取了哪些互动；第三类科学家关注的问题是，怎样才能设计出更好的软件界面或外设，使之满足用户的所有使用需求。

人机交互领域的重要性日益凸显。安·布兰福德是伦敦大学学院教授，专攻人机交互。她表示：“在计算科学的早期发展阶段，工程师包办了所有的设计工作。他们注重的是动手制造机器，并探索其中的原理。人机交互领域的科学家则不一样，他们关心的是怎样提升用户的体验，或者说，为什么机器给用户带来的体验不够好。直到今天，这两派科学家之间的矛盾依然存在。”

遗憾的是，在计算机科学领域，不如人意的设计比比皆是。有时候，之所以会出现这种问题，是因为一些不合时宜的旧方法没有被及时淘汰。比如，靠密码验证身份的方法虽已过时，但依然得到了日益广泛的采纳。安吉拉·萨斯表示：“密码验证法最开始问世时，人们一般只需要设一个密码，很容易就能记住。但是这种方法通行之后，人们开始为所有的账号都设置密码，记忆负担太大。密码根本不适合这样用。毕竟，人的记忆力有限，不可能把所有想记的东西都记住。很多事情如果不经常回想，很快就会忘记，如果不借助任何提示，是很难回想起来的。相似的信息在大脑中的存储位置也是相同的，它们会相互竞争，当你不再需要某个密码的时候，你不能把它直接清除掉，它会留在那里挤占你的记忆空间。所以，如果密码的数量超过了两、三个，

而且又不经常使用，我们就根本记不住。”

但是，安全系统工程师依然乐此不疲。他们一味地帮助用户改进密码质量，却从来没有把“人”的因素考虑进去。他们经常制定密码政策，以确保用户采用的是安全的密码。（譬如密码的长度有最低限制，必须采用大小写字母、数字及标点符号混合的密码，而且最好每过几周更换一次。）“我们做过研究，受试者花了20分钟的时间才想出一个符合政策的密码，”萨斯表示，“如果把一个人每天花在密码上的时间累加起来，就会得到一个惊人的数字。”

萨斯与好几家公司合作开展了上述研究。她以分钟为单位，监控员工在上班时使用计算机的情况。有时候，员工需要登录许多不同的计算机系统，他们在这上面浪费了大量的时间。“员工每天单是用来登录系统的时间加起来就有将近30分钟了，”她说，“一年下来就是三个星期！我跟公司的人说，‘你知道员工一年单是为了登录不同的系统就要花掉三个星期吗？’他们都吓了一跳，‘不会吧！这不可能！’我把数据给他们看。数据就摆在那里，这种事情绝对是可能的，但确实是太疯狂了。”

一旦计算机系统的设计师认为软件特性比用户体验更重要时，就会带来一个新的问题。软件的早期版本可能带有精心设计的交互界面，功能

也是一目了然，后来的版本总要加点新的东西。所以越来越多的新特性被加了进来。选择多了，功能多了。最常见的结果就是，本来好端端的软件被弄得臃肿不堪，而且过于复杂，不仅运行速度变慢，而且学习和使用起来都很不方便。“数以百万计的用户都被这种软件弄得苦不堪言。”萨斯说。

“问题是，设计这个系统的人根本就没有怎么考虑过别人会怎么使用它，”布兰福德说，“操作效率、使用效果和用户体验原本应该是不容忽视的因素。产品设计的好坏事关重大，常用的产品在设计上就更应该精益求精。设计得不好，用户必然一肚子火。如果产品对人身安全很重要，那可就是关系到生死的问题了。界面设计得差，那就意味着做同样的事情，用户花费的时间会比原本多得多，因此也就更容易犯错误。在医院里，医护人员如果操作仪器的时间比看护病人的时间还长，那就有问题了。”

不过，时代在变化。很多企业已不再将系统自动升级到最新版本了。毕竟，如果当前的系统使用起来得心应手，何必多此一举，花费时间和金钱培训员工学习新技术呢？与此同时，人机互动领域的科学家也提出、并验证了不少改善用户体验的方法。“体验实验室”（Living Laboratory）就是一大创新，新技术发布前，用户就可以在这

里体验并提出意见。这个创新的思路和药物临床试验非常相似——首先，新药需要在实验室条件下仔细评估，确定它们的功效正常，而且没有副作用。只有经过足够多的严格测试，药物才能上市。体验实验室就是计算机系统接受“临床试验”的地方。

已经有一些互联网公司在运用类似的原则来测试它们的新版网页了。“大网站通常会做A/B测试，”布兰福德表示，“他们会把测试版网页上线。访问原网页的用户当中，有十分之一会被重定向到测试版页面上。公司会比较两个用户群的行为和反应，然后据此决定应该采用新版网页还是旧版网页。当然，你需要有足够的流量支持，这样才能拿到具有统计学意义的结果。”目前，谷歌、微软、Facebook、易趣（eBay）等家喻户晓的企业都已采用这一策略。

眼动追踪（eye-tracking）是另一项卓有成效的人机交互技术。人在使用计算机界面时，计算机监测人的眼睛在屏幕上的注视点。研究人员可以借此分析界面是否太乱、是否具有吸引力、哪些部分未能吸引用户的目光。“富媒体⁽¹⁰⁾

（rich media）既有可能吸引眼球，也有可能干扰视线，”萨斯说，“人们浏览界面一般都是有目的的，他们会下意识地跳过某些无关紧要的信息，这样就在界面上造成了阅读盲点。”

通过眼动追踪技术，研究人员很快就能知道，怎样才能在最短时间内给用户提供最有针对性的信息，或许还可以根据用户在界面上的注视点，动态地调整界面上显示的内容。“只要将智能界面和自适应技术⁽¹¹⁾结合起来，我们就能使用户在不受干扰的情况下快速找到自己想要的信息。”萨斯表示。

谷歌等大企业已经率先在智能界面上投入重金，以帮助用户更快地检索信息。“搜索引擎会寻找有效的算法，以检索与关键词相关的介绍性文章，这样一来，用户不需要依靠高级搜索，就可以很快地找到不错的背景资料。如果用户可能输入了错误的关键词，搜索引擎会根据需要，给用户相应的提示。搜索引擎还会在结果网页的标题下方显示一段与关键词匹配的文字摘要。这些事情谷歌基本上都做到了。”布兰福德表示。

然而，一旦眼动追踪等尖端技术在个人计算机上普及开来，又有可能产生新的问题。广告商不会放过任何一个了解潜在客户的机会。事实上，很多广告商早已把我们的购买习惯记录在案，甚至还会监视我们在网页上点击鼠标的位置，以方便他们精细地调整广告投放的位置。如果他们连我们在屏幕上注视的地方都摸得一清二楚，说不定会采用一些令人生厌的手段来施加影响。同样是收集用户数据，如果其目的是为了改

善人机互动的体验，那么这样的做法是喜闻乐见的；如果其目的单纯是为了牟利，那就是另一回事了。“问题并不在于广告商想定点投放广告，”萨斯说。“问题在于，广告商在诱导我们花不想花的钱，买根本买不起的东西。这样还不如开发一个系统，直截了当地告诉广告商我们买得起多贵的东西；或者告诉他们，这个月没钱了，别再拿那些广告来烦我们。”

随着智能界面的不断改进，人们开始担心总有一天，计算机会了解人的深层次需求，并据此开出难以抗拒的诱惑条件。安吉拉·萨斯曾经在一次研究中听受试者谈到了这个问题。受试者说：“我担心人脸识别技术普及以后，我们连闲逛都不得安生了。比方说你有半个小时的时间不知道怎么打发，于是进了一家商店随便看看。其实也没什么衣服要买，根本不需要消费。这时候你的手机突然响了，来了一条短信说，‘您好，苏西，欢迎回到大卖场。我们注意到，您刚才在这些裤子旁边多停留了一点时间。根据零售心理学理论，我们知道您应该对它们有兴趣。所以如果您愿意在五分钟内买下两条裤子的话，我们愿意给您八折优惠。’”

想要解决人机交互的问题着实不易。只有给计算机提供更加详实的信息，才能使它们更好地服务于人，甚至造福于人。但这样做也会带来隐

私泄露的问题。

不过，近年来，以顾客为导向的计算设备显然走对了路子。它们的设计通常比较简洁大方，会刻意减少一些累赘的功能，操控起来也更加方便自如。与过去布满按键的手机相比，现在的手机普遍采用触屏设计，用户只要用手指轻触和划动显示屏，就可以流畅自如地体验手机的各种功能。“要想设计出伟大的作品，往往需要尽量减少累赘，追求简约、利落的效果，”萨斯表示，“而且，在功能的设计上，要把握好一个度，有些事情机器可以为用户代劳，但有些事情不可以越俎代庖。”

计算机：人的宠物

人机交互领域的研究人员还需要面对一个问题，那就是众口难调，毕竟每个人的好恶不一样。事实上，安东尼奥·达马西奥（Antonio Damasio）等神经科学家认为，我们的决策能力在很大程度上受到情感影响。达马西奥引用了一个著名的案例：铁路工人菲尼亚斯·盖奇（Phineas Gage）遭遇了工伤事故，一根大铁杆击穿了他的头骨，损坏了大脑左前额叶部分。盖奇奇迹般地活了下来，但是他的性格发生了很大的变化。从此以后，他失去了情感能力，而且不能做出任何理性的决策。从这样的案例中，达马西

奥归纳出了一个情感模型，模型显示，人们在决策过程中会把情感当做某种捷径使用。如果过去的一项决策让我们更快乐，那么在未来遇到相似的情形时，我们就更倾向于做出相似的决策。如果过去的某个决策让我们感到不快，那么再遇到类似的情形时，我们的选择可能就会与以往不同。所以说，情感使决策过程更加迅速，只要借助情感进行决策，人们就不会劳心费力地分析每个决策的利弊，以至于犹豫不决。“人们在决策时会梳理过去的经验，并与现状进行参照。有了过去的情感体验，人们会以更加灵活的方式进行决策，”娜迪亚·贝尔托说，“情感能够从根本上影响人们决策及分配注意力的过程。可以说，如果没有情感的参与，我们绝对无法采取明智的行动。”

情感计算是人机交互领域的新分支，达马西奥的模型在该领域很受重视。情感计算的目标是创造出一种新技术，这种新技术可以感知人类情感状态，然后对收集的情感数据进行处理，进而提高计算机的性能。在理想情况下，计算机还可以对人类的情感做出反应，甚至可以表达自己的情感。

情感计算着眼于用户在与新界面和设备互动时的体验——新的界面和设备能否给用户带来愉悦的体验？能否增进用户的心理健康？研究人员

还采用了机器学习技术，使计算机能够分析人的面部表情、肢体语言、语音语调，进而判断人类的情绪状态。他们还与人工智能领域的研究人员合作，开发出了具备“个性”和“情感”的计算机和机器人。

情感计算技术除了可以应用到玩具上，还可以为很多人解决实实在在的问题。“目前我们正在开发一项技术，帮助慢性疼痛患者进行自主理疗，”贝尔托说，“开发这些技术的目的，是为了激发他们主动参与体育运动，让他们多活动活动筋骨。正常情况下，患者在医院做完一个治疗周期后，就出院了。但这仅仅只是一个月的理疗而已，远远不足以治愈慢性疼痛。可是患者出院后就不再主动锻炼身体了。恐惧和焦虑会使患者产生疼痛的错觉，进而使他们不敢运动。但是不运动恰恰是他们应该极力避免的选择，因为这会使身体状况进一步恶化。”

即便身体已经康复，伤口早已愈合，很多病人还是要继续忍受慢性疼痛带来的折磨。负面情绪与肉体疼痛交织在一起，让病人觉得这疼痛的感觉无比真实。贝尔托之所以开发理疗技术，是为了取得两个目的：一是帮助患者改善身体状况，并把治疗取得的进展反馈给他们。二是为患者驱除心理阴影，鼓励他们振作起来，帮助他们在锻炼之前放松身心。此外，为了掌握病人真实

的心理状态，研究人员采用了计算机算法分析病人的动作和面部表情。“沮丧、焦虑和恐惧都会与疼痛如影随形，”贝尔托说，“所以我们要看看，情绪状态是否会影响患者的身体动作。我们观察到，患者为了避免感觉到疼痛，会做出许多保护性的动作。但如果他们不害怕的话，身体的动作就会放松许多。人在紧张的状态下，身体肌肉的调动方式与放松的状态下截然不同。”

有趣的是，一项技术是否通人性，直接决定了人们对它的态度。如果当前的技术能够更通人性，甚至像小猫、小狗等宠物一样，能给人情感上的寄托，那么人们的反应也会迥然不同。如今的计算机或机器人除了用于护理病人，还可以推而广之，用于照顾小孩和老人等等。“几年前，找机器人陪护还是一件无法想象的事情，”贝尔托说，“现在，这些都可以实现了。研究人员曾将机器人宠物引入养老院，观察老人的反应。”结果令人惊讶，为数不少的研究表明，机器人宠物很受欢迎。“我觉得社会已经准备好接受这些新鲜玩意了，”贝尔托说，“工业界也在耐心观察，机器人宠物的市场肯定不小。所以我觉得很快这项应用就能推广开来。”

人机一体化

在持传统观点的人眼里，计算机只是一件没

有感情的机器设备而已，尽管它能与我们互动，也能影响我们的喜怒哀乐。或许人们对计算机的情感往往会是沮丧或者恼怒，毕竟不管是谁，碰到设计差劲的应用，想必都会大感不快。但是退一步想，其实计算机有很多时候并不是那么的令人讨厌。多亏了以恩格尔巴特为代表的先驱科学家，现代平板计算机等消费类电子设备的设计和使用界面才会做得越来越好，人们从中得到的乐趣也越来越多。正因为苏泽兰及其学生的开创性成果，如今的电影才可以完全脱离胶片的束缚，实现数字制作，而且效果与传统电影一样令人满意。除此之外，计算机游戏也给玩家带来了丰富的情感体验；即时通讯软件使人们能够与千里之外的亲友实时沟通联系；有些机器人甚至已被用于照料或陪护自闭症患儿和孤寡老人。

或许更令人惊讶的是，计算机科学正在与社会科学相融合，形成了一门交叉学科，这门学科以人为中心，着眼于人类如何与技术互动、怎样才能做出令人满意的人机交互界面等问题。虚拟现实和计算机图形学不仅可以用于制作精美直观的图片、电影和演示动画，还可以用来治疗患有各类恐惧症的病人、培养和提高人们在新环境下的应变能力。用户界面的设计讲究的是简洁大方，以人为本，这样才能使技术真正惠及大众，满足人们的需求。正如我们所见，广告商已经开

始利用人机交互技术，监视消费者的一举一动，以确保广告能够投放在最合适的位置。

未来的计算机界面会发展到什么地步，我们虽然很难预测，但不妨张开想象的翅膀，大胆猜测一番。或许终有一天，人们只要轻轻一挥手，就可以向计算机发送指令；人机界面会发展到与人心有灵犀的地步；虚拟现实或增强现实眼镜将与普通眼镜一样稀松平常；机器人宠物将拥有自己的个性和情感，像小猫、小狗一样与人们互动；计算机将成为人们生活的一部分，以至于“界面”这个词都显得累赘。

我们不妨再设想得远一点，思考一个有趣的问题：未来的计算机会不会因为受了委屈而变得沮丧甚至愤怒呢？

注释

[\(1\)](#) 两指捏拢或分开即可实现屏幕的缩小或放大。

[\(2\)](#) 可将图片旋转到任意角度。

[\(3\)](#) 可将文档页面上移或下移。

[\(4\)](#) 计算机图形学（Computer Graphics，简称CG）是一种使用数学算法将二维或三维图形转化为计算机显示器的栅格形式的科学。

[\(5\)](#) 这种方法可对3D模型各顶点的颜色进行平滑、融合处理，将每个多边形上的每个点赋以一组色调值，同时将多边形着上较为顺滑的渐变色，使其外

观具有更强烈的实时感和立体动感，不过其着色速度比平面着色慢得多但是效果要好得多。

[6] 在单线程方式下，计算机是一台严格意义上的冯·诺依曼式机器，一段代码调用另一段代码时，只能采用同步调用，必须等待这段代码执行完返回结果后，调用方才能继续往下执行。有了多线程的支持，可以采用异步调用，调用方和被调方可以属于两个不同的线程，调用方启动被调方线程后，不等对方返回结果就继续执行后续代码。被调方执行完毕后，通过某种手段通知调用方：结果已经出来，请酌情处理。

[7] 矢量图使用直线和曲线来描述图形，这些图形的元素是一些点、线、矩形、多边形、圆和弧线等等，它们都是通过数学公式计算获得的。例如一幅花的矢量图形实际上是由线段形成外框轮廓，由外框的颜色以及外框所封闭的颜色决定花显示出的颜色。

[8] 运动捕捉系统是一种用于准确测量运动物体在三维空间运动状况的高技术设备。它基于计算机图形学原理，通过排布在空间中的数个视频捕捉设备将运动物体（跟踪器）的运动状况以图像的形式记录下来，然后使用计算机对该图象数据进行处理，得到不同时间计量单位上不同物体（跟踪器）的空间坐标（X，Y，Z）。

[9] 情感计算（Affective Computing）就是要赋予计算机类似于人一样的观察、理解和生成各种情感特征的能力，最终使计算机像人一样能进行自然、亲切和生动的交互。情感计算是人工智能领域的下一个突破，可能在于赋予计算机情感能力。

[10] 富媒体是指使用浏览器插件或其他脚本语言、Java语言等编写的具有复杂视觉效果和交互功能

的网络广告。

[\(11\)](#) 自适应技术的概念描述了一类新的“智能组件 / 智能结构”，这些智能组件在工作条件改变的时候具有一个主动适应和目标优化调节的机械特性。这种主动的组件满足了改善系统机械特性、效率、性能及其他特性的要求。

构建仿生学大脑

自计算机科学诞生以来，研究人员在私底下一直自视为人工大脑的构建者。毕竟，既然人类能够凭借神经元释放的数十亿个电脉冲产生思维活动，计算机为什么不能如法炮制，利用电子电路释放的电脉冲来产生思维活动，进而完成各种各样的智能任务呢？随着科学技术的日新月异，或许终有一天，计算机不仅能够学习知识、预测未来、听说读写、自由行走，还可以诊断疾病、驾驶汽车、探索宇宙。

但即便如此，人工智能的界定标准是什么呢？是逻辑推理能力？抑或是从日常生活中吸取经验的能力？一旦拥有了智能，计算机将如何看待自身乃至周围的环境？人类真的能够制造出具有自我意识的人工大脑吗？

伴随着欢快的背景音乐，一名身材高大、但略显消瘦的美国男子出现在粗糙的彩色短片中，他身穿黑色西服，右手举着一样东西说道：“它叫忒修斯（Theseus）。”

短片随即切换到一个特写，一只白色小老鼠身处迷宫之中，它不停地向前跑，时而东窜窜，西窜窜。“忒修斯是一只电子控制的老鼠，它有能力运用试误法⁽¹⁾解决某些问题，然后记住解决方法。换言之，它可以从经验中学习。”

克劳德·香农又一次做出了举世瞩目的成果。在第八届控制论⁽²⁾大会上，他展示了自己发明

的精巧机器，引起了在场科学家的一片惊叹与赞誉。也许想要听起来更严肃一点，他在科学会议上通常喜欢把自己发明的机器老鼠称为“手指”。

“你看，现在‘手指’正在探索迷宫，寻找终点，”香农在会上一遍演示机器老鼠，一边介绍道，“每次它走到方块中央时，就会做出新的决定，明确下一步的走法。如果“手指”碰到了迷宫墙，马达就会倒转，“手指”就会回到方块中央，然后选择一个新的前进方向。这个选择是根据前面获得的知识和某些策略做出的，其中的策略解释起来稍微有点复杂。”

“这里边有策略？”数学家沃尔特·皮茨（Walter Pitts）问道，“不是随机的？”

“完全没有任何随机因素，”香农答道，“一开始我也想过加入概率因素，但是后来觉得如果用的话，难度比较大，用固定的策略会简单一些。”香农回到了自己演示上。“现在大家可以看到，‘手指’的探险结束了，它已经到达了终点，马达也停止了运转，“手指”上的小信号灯自动点亮并发出铃声。走迷宫任务完成。”

“既然它知道怎么从起点走到终点，”计算机工程师朱利安·毕格罗（Julian Bigelow）问道，“那它知不知道怎么原路返回？”

“它不知道。你可以看看这个矢量场，‘手指’所走的每一步都对应于一个矢量。从图上可以

看出，这些矢量的方向都是各异的，但是如果从反方向逆推，就能看到一些分叉点，所以‘手指’没有办法原路返回。”香农答道。

“这就好比一个人对某个地方很熟悉，可以随便溜达，但不会特意去记路线。”神经科学家、控制论学者沃伦·麦卡洛克说着，陷入了沉思。

生物物理学家、控制论学者冯·福尔斯特（Van Foerster）问道：“如果没有终点会怎么样呢？”

“如果不设终点，机器就会逐渐摸索出一条路径。它会将每一个方格和死胡同都走一遍，如果找不到出路，就会沿着这条路径一直走下去，每走一步都会试图寻找出路。”

“这和人类也太像了，”社会科学家劳伦斯·弗兰克（Lawrence Frank）惊叹道。

“我想到了《1984》的作者乔治·奥威尔，他生前要是看过这台机器就好了。”精神病学家亨利·布罗欣（Henry Brosin）说道。

教计算机下棋

香农的走迷宫机器是用电话开关继电器还有马达磁铁制成的。后者的作用是驱动机器老鼠（即“手指”）在迷宫里移动。尽管这个简单的机器人拥有基本的决策能力和记忆力。它的控制机

构还不算是通常意义上的计算机。然而，它的继电器已经足以满足“手指”探索迷宫并记忆路线的需要。后来人们模仿香农的机器老鼠，做出了几款仿制品，直到20世纪70年代，还有科学家在公开演示这类仿制品。“手指”的发明是一项了不起的成就，香农作为创立了布尔逻辑、信息论和通信理论的先行者，能够做出如此精巧的机器，着实令人惊叹。启发香农发明“手指”的，是另一款同样令人不可思议的机器，那就是会下棋的计算机。

1950年，也就是香农（如图7）发表《通信的数学理论》之后的第二年，他就又发表了一篇论文，讨论了会下棋的机器。1951年，他研制出了会走迷宫的计算机。



图7 克劳德·香农和会走迷宫的机器老鼠“忒修斯”，“忒修斯”由磁铁驱动。1952年摄。

香农对国际象棋的理解方式很是特别。约翰·冯·诺依曼与经济学家奥斯卡·摩根斯特恩（Oskar Morgenstern）在1946年合写的巨著《博弈论与经济行为》（*Games and Economic Behavior*）对他的影响很深。他把国际象棋想象成类似于老鼠在迷宫中寻找路径的问题。棋手在轮到自己走棋时，他所思考的，无非是步步为营，对手于置于死地，这就好比在一个迷宫里探索，起点是当前的落子位置，终点是致胜的棋步。正如陷在迷宫

里的老鼠一般，有些棋路能走，有些则不行。棋盘上虽然没有墙挡住棋子的去路，但是行棋依然需要遵守规则，只能按照某个确定的方向移动棋子。比如这一步走马，可能会给下一步走车打开通路，同样也有可能的是，这一步走了兵，把车的进路给挡住了，这样就影响了下一步走车的行动。每一步可以有多种选择。每做出一个决定，未来又会有新的棋路可供选择。怎样看待走棋的问题很有讲究，除了将走棋看做是走迷宫，我们也可以把眼光扩展到整个棋盘，而不是拘泥于一颗棋子的动向，这样，每步棋都可以视为棋盘表面状态的一次变化。

但是，与老鼠尝试在静态迷宫中寻找路径不同，在国际象棋里，棋手采取的路数还取决于对手的棋路。所以，要想寻找制胜的最佳棋路，必须假设对手的走法与自己设想的完全一致。这就好比迷宫里的老鼠有一个对手，它会不停移动墙壁以阻止老鼠走到目的地。

冯·诺依曼在研究经济学和博弈论时，就已经预见到了这个问题。他证明了最小最大定理

（minimax），该定理表明，博弈双方的任何一方都会采取最有利于自己、而最不利于对方的策略。香农在会下棋的计算机中应用了这一思想。计算机在走棋前会考虑自己和对手接下来会走哪两步棋，然后用一个棋局评估函数来分析可能造

成的局面。该函数由一系列经验法则（或启发式搜索[\(3\)](#)算法）组成，用于判断己方和对手哪一方更占优势，占据的优势有多大。

在国际象棋中，对弈双方每走四手棋（双方各走两步）之后，就会形成新的局面，从中可以看出哪一方更占优势。所以计算机的目标就是，找出接下来可能是最高明的两步棋，尽量增加自己的胜率，同时压低对手的胜率——当然，前提是，对手的走法与计算机的设想完全一致。一旦确定了哪两步棋可能会是最佳的选择，计算机就会走出第一步，然后观察对手的反应。对它来说，最坏的情况莫过于对手走出最强应招，但这也符合它的预期。最好的情况自然是对手不慎失策，下了一手昏招。这样一来，场上的局面或许会比它一开始预测的还要好。于是，计算机接着分析下面该走的两手棋，直到决出胜负。

按照香农的设想制造出来的计算机不可能称得上是完美的棋手。但是香农的“最小最大算法”为日后无数的计算机博弈系统打下了理论基础。这类计算机每次落子前都会采用搜索算法[\(4\)](#)，从数量庞大的选择当中寻找出最优的走法。此外，香农取得的突破还在于，他发现了两种类型的博弈算法，一是暴力搜索，穷举双方在每一步可能采取的所有招法，从中找出最好的；二是上文所说的启发式搜索，只评估最有可能致

胜的招法。香农更偏爱后一种算法，因为它的每一步运算量更少，因而可以运算更多的步数，起到料敌先机的效果。

从20世纪40年代末到50年代初，除了香农以外，还有其他科学家在思考如何研制计算机博弈系统的问题。艾伦·图灵（在纸上）编写了第一款智能国际象棋程序。它的原理与搜索算法一样，都是从一系列的可能性中筛选出最佳的棋招。那时候，还没有一台通用计算机可以运行这款程序，因此图灵只好自己扮演计算机的角色，按照自己写出来的程序和同事对弈，每走一步都花了半个小时的时间，结果他输了。

或许正是由于这些先驱的大胆尝试，计算机在国际象棋乃至国际跳棋、围棋等博弈游戏中的表现，已成为其智能化程度的一个标杆。在未来的几十年里，为数众多的研究者还将继续探索如何提高计算机棋艺的方法。以国际象棋为例，1997年，IBM公司制造的计算机“深蓝”击败了国际象棋大师卡斯帕罗夫（Kasparov）。这是计算机有史以来第一次击败现役的国际象棋世界冠军，而且比赛是严格按照标准国际象棋锦标赛的规则进行的。这是人工智能研究史上的标志性成就。虽然作为一件杰出的工程学作品，“深蓝”在下棋时已经表现出某些类似于人类棋手的卓越能力，但它终究是一台机器，无论是开局还是收

尾，都必须依赖不计其数的棋谱组成的庞大数据库。所谓的“智能”特性，大多只是纯粹的暴力搜索罢了。毕竟，计算机可以在每秒内分析和评估数十亿种可能的棋招。“深蓝”（乃至迄今为止的大多数计算机博弈系统）充其量不过是巨型的棋招搜索机器罢了，没有学习能力，更谈不上成长和进步。

人工智能的诞生

1950年，香农和图灵都以会下棋的计算机为例，对机器智能进行了探讨。香农认为，如果计算机可以下棋，那它或许也可以做逻辑推理、做翻译、甚至作曲。图灵也认为，凡是人类智能可以完成的壮举，会下棋的计算机或许也能完成。

图灵不像香农那么擅长制造设备，制作机器老鼠并不是他的风格，他更加注重的是理论思辨。不久前，图灵撰写了一篇论文，题为《计算机器与智能》（Computer Machinery and Intelligence），文中阐述了他对机器智能的看法。这篇论文为计算机科学的诞生奠定了一大基础。图灵在开篇写到，“我建议大家思考一个问题——‘机器会思考吗？’”。

为了解决这个艰深的问题，图灵提出了“模拟游戏”（imitation game）的概念（今人称之为“图灵测试”）。他认为，这是判断计算机有没

有智能的科学方法。判断的依据并不在于计算机会不会下棋、会不会走迷宫，而在于它会不会与人沟通交流。

图灵所描述的模拟游戏是这样的：假设两个房间里分别有一个人、一台计算机，你作为提问者进入第三个房间，通过一些装置（如键盘）对他们随便提问。假如过了五分钟，你还判断不出谁是人、谁是计算机，那么这台计算机就通过了图灵测试。

图灵在论文中表示：“我确信，在五十年之后，计算机的储存能力将到达 10^9 次方级别，在次基础上对计算机进行编程，就可以使它们玩模仿游戏的水平大幅提高，足以让一般的提问人在五分钟的讯问后，猜中交谈对象身份的机率不超过70%……我相信，到了20世纪末，学术界以及大众的主流观点会发生巨大的变化，就算有人提出机器会思考的观点，也不会有人存疑。”

图灵并不是英国唯一一个研究智能计算机的人。他所属的比例俱乐部（Ratio Club）也对此非常热衷。比例俱乐部是由英国的一群年轻科学家在不久前新建而成，其宗旨是探讨两个问题：一是行为产生的机制，二是人脑和机器的信息处理机制。该俱乐部经常邀请神沃伦·麦卡洛克

（Warren McCulloch）等美国科学家参与讨论活动。

20世纪50年代，正当英国人忙于思考探索智能计算机的有无问题时，在大洋彼岸的美国，研究人员从这个问题出发，开创了一个全新的领域。当时美国召开了一系列控制论会议（香农正是其中一场会议上展示了自己发明的机器老鼠），不少科学家开始积极致力于提高计算机的智能化程度。1956年，这些志同道合的人在新罕布什尔州的达特茅斯学院（Dartmouth College）召开了为期一个月的集体研讨会。大会的组织者包括克劳德·香农、约翰·麦卡锡（John McCarthy）、马文·明斯基和纳撒尼尔·罗切斯特（Nathaniel Rochester）。与会者包括雷·所罗门诺夫（Ray Solomonoff）、奥利弗·赛尔弗里奇（Oliver Selfridge）、特雷查得·摩尔（Trechard More）、阿瑟·塞缪尔（Arthur Samuel）、赫伯特·西蒙（Herbert Simon）和艾伦·纽厄尔（Allen Newell）。这些人后来都成为了人工智能领域的先驱，“人工智能”这个术语也是在大会上确定的。

在人工智能界，此次会议可谓开天辟地的大事件。也正是在这次会议上，研究人员确定了很多日后需要解决的关键性问题。这些问题即使放到今天，也依然是研究的重点。比如：英文等自然语言能否用于编程？可模拟人脑神经元的程序能否编写出来？计算机能否通过学习而不断进

步？计算机应该如何表达信息？在算法中增加随机性，是否就能提高创造力？

1946年，摩尔学院开办的讲座在全世界范围内掀起了一轮研制计算机的热潮，时隔近十年，达特茅斯学院召开的集体研讨会催生了人工智能这个全新的研究领域，这一领域势将改变整个计算机科学的面貌。此后，新的人工智能技术层出不穷，并在计算机科学的各个领域大行其道。“我觉得人工智能领域的先驱开创了许多高级的计算技术，比如非数值运算⁽⁵⁾程序设计、汇编语言、硬件描述语言⁽⁶⁾等等，”苏泽兰表示，“这些成果已经深入到了计算机科学的基础领域，现在连本科生都需要掌握这些技术了。”

然而，人工智能的发展还远远称不上是一帆风顺。从很早的时候开始，科学家就开始为计算机智能的发展方向争论不休。他们分成三派，第一派以麦卡锡为代表，认为逻辑推理是计算机智能化的必由之路。第二派以麦卡洛克、皮茨为代表，认为构建神经元模型才是正道。第三派以明斯基为代表，认为计算机要想实现智能化，就必须首先具备现实世界的知识。三派科学家在唇枪舌剑的争论中掀起了几轮炒作的热潮，此后，炒作谎言的破败使公众和投资界对人工智能的信心崩溃，人工智能领域几近衰落⁽⁷⁾。在这一时期，明斯基正是站在风口浪尖上的关键人物。

人工智能的兴起

马文·明斯基1927年出生于纽约。他还有一个姐姐和一个妹妹。父亲是眼科医生，但儿时的明斯基似乎对化学和新兴的电子学更感兴趣。他从小就在私立学校接受教育，而且表现非常出色，天分显露无疑。五岁时他参加了一次智力测试，答案超乎年龄的优秀。有一次，小马文碰到了这样一道题。假如皮球滚入了一望无际的高草地中，怎样才能把它找出来？他的老师们一时半会儿也没摸着头绪，觉得似乎应该从草地中间开始寻找皮球，然后一圈一圈的向外搜索。小马文马上意识到这种方法存在缺陷，他立刻反驳了老师们答案。因为如果你想从草地中间开始搜索的话，就必须首先穿过草地到达中央起点，所以，从外向内搜索显然要更有效率。马文至今对这件事情记忆犹新，他说：“那时候，我第一次发现大人并不是完美的，这种孩提时代的觉醒经历想必每个人都不会忘记。”

高中毕业后，明斯基在美国海军服役了一年，随后进入哈佛大学读书。他一开始选择的是物理专业，但没过多久就对遗传学、数学、音乐和人工智能产生了浓厚的兴趣，不过他对主流理论并不认同。明斯基还经常跑到哈佛大学心理学系和生物系的实验室里做实验（比如解剖小龙虾，观察它们的神经元）。尽管他后来转攻数学

专业，撰写了一篇有关拓扑学的论文，但做实验对他来说，一直是校园生活的重要组成部分。

也就是在这段时间，明斯基逐渐被神经生理学家沃伦·麦卡洛克和数学家沃尔特·皮茨的工作所吸引。1943年，麦卡洛克和皮茨发表了一篇文章，阐述了一个抽象的神经元模型。这个模型把真实神经元的工作方式在极大程度上进行了简化：每个人工神经元都会接受几个输入信号（用二进制表示，1代表输入，0代表没有输入），总输入信号是这几个信号的加权（乘上1或是-1）再求和。然后再把计算结果与一个阈值相对照。如果高于阈值，人工神经元就能产生一个单路输出，记为二进制里的1，低于阈值就没有输出值，记为0。如果把这些单独的人工神经元用正确的方法连接起来，即上游神经元的输出值可以作为下游神经元的输入值的话，即便这些神经元的功能极为简单，它们也可以像简单的逻辑电路一样工作。

对于如何构建神经元模型，明斯基有着自己的看法。“我想象中的大脑是由无数的小型继电器构成，这些继电器就是神经元，每个神经元是否发送神经电脉冲信号还要受到概率因素的影响。这个设想用现在的术语来表达就是‘随机性神经网络’。”按照明斯基的观点，神经元可以通过调整不同输入信号的权重比来进行学习。这与唐

纳德·赫布（Donald Hebb）不久后在论文中提出的思想——即“赫布型学习”（Hebbian learning）不谋而合，赫布型学习的理论，用一句话可以总结为：如果两个神经元同时被激发，它们之间的联系就会强化⁽⁸⁾。

后来，明斯基前往普林斯顿大学攻读博士学位。没过多久，他就开始对实际建造一个神经网络产生了兴趣。在哈佛大学一位同事的帮助下，他拿到了项目基金，并在1951年造出了原型机。这是一台装有三百根真空管的庞然大物，它能像大脑一样进行学习。为了测试原型机的能力，研究人员设计了类似于老鼠走迷宫的问题。令人惊讶的是，原型机不仅摸索出了一只老鼠走迷宫的巧妙路线（一旦在某个地方做出了正确的选择，之后遇到相同的情况也会更会更容易做出决策，这样反过来又会强化正确的决策过程），还可以同时操纵两只老鼠寻找路线。后来，明斯基将这个项目作为博士学位论文的研究课题，继续开展研究。他想知道如果再加一个存储器，原型机的人工神经网络会如何预测特定的决策造成的后果。“如果一台机器或动物需要对新的情况做出反应，它们会从记忆中搜索已有的经验，据此判断特定的反应会造成什么后果，”明斯基说，“如果曾经做出的某个选择带来了不好的结果，那么机器就会选择其他的反应方式。”

明斯基没有使用当时已有的计算机。“一方面我有点犯怵，觉得它们太复杂了，”明斯基说，“另一方面，我觉得它们还不够强大，不具备优秀的学习能力。不管怎么说，我的论文研究的是，怎样才能让人工神经网络学会学习。”冯·诺依曼是答辩委员会的成员，所以明斯基时不时就会找他讨论自己的论文。当时，系主任阿尔伯特·塔克（Albert Tucker）对于明斯基的研究工作还不太放心。“塔克后来跟我说，他曾经拿着我的论文去找冯·诺依曼征求意见，他说，‘这篇文章看起来挺不错的，但是我有点拿不准，不知道它讨论的问题属不属于数学范畴。’冯·诺依曼回答他说，‘好吧，就算现在不是，以后保不准就是了。我们还是多鼓励一下年轻人，让他好好做完吧。’于是我就拿到博士学位了。”

1954年，明斯基拿到博士学位后，受到冯·诺依曼、香农和维纳（Weiner）的一致支持，担任了哈佛大学的研究员。在接下来两年的时间里，他深入探讨了人类和机器智能的问题，形成了自己的思想。1956年，明斯基协助组织了达特茅斯研讨会。就算是在开会，他也没有丝毫放松，依然在思考人类和机器智能的问题，比如怎样才能设计一款自动证明定理的程序，让它证明欧几里得的部分定理。两年后，明斯基加入了麻省理工学院的香农课题组，与麦卡洛克、麦卡锡等人共

事。

在那个年代，技术的发展日新月异，创新的成果层出不穷。计算机先驱相继提出了一些非常重要的概念，如分时⁽⁹⁾（timesharing）。人工智能的理论得到进一步发展，研究人员开始从不同的角度审视人工智能问题。“怎样才能制造一台会思考的全能机器？或者进一步说，就算制造出了一台全能的机器，你能把它解决问题的运算过程称为思考吗？所以这是我们要解决的第一个问题，”明斯基说，“第二个问题是，有没有什么指导性的理论可以界定机器能否思考的问题？人们能否弄清楚智能的本质、原则和一系列其他的奥秘？这两个问题其实涉及到对人工智能的本质定义，问题是它们太抽象了，没什么实际意义。通俗地说，人工智能实验室其实并不神秘，研究人员所做的，无非就是捣鼓机器，想方设法地让它们做一些当前的技术难以实现的事情。所以，从这个意义上讲，人工智能其实就是计算机科学中最前沿，更需要前瞻性的一小部分领域。”

20世纪60年代初，科学家们提出了一些比较明晰的人工智能设计思想，他们分成几个不同的派别，彼此之间水火不容，每一派的理论看起来都大有前途。约翰·麦卡锡是明斯基在麻省理工学院的同事，他认为，要实现机器智能，必须依靠数理逻辑。那时候，语言学家艾弗拉姆·诺姆·乔

姆斯基（Avram Noam Chomsky）风头正劲，他发现人类语言的结构（句法）与和它要表达的意思（语义）有差异。受他的影响，麦卡锡一直致力于用逻辑和逻辑化的计算机语言使计算机学会推理思考。他的工作也为某些计算机语言的开发铺平了道路，比如列表处理语言LISP，还有后来的PROLOGo（一种逻辑编程语言）。麦卡锡对人工智能领域的发展影响深远，时至今日依然如此。“麦卡锡的思路就是要让计算机像人类一样推理思考，”罗宾·赫希表示，“不管你想做什么，你必须首先设计一套正确的逻辑，把它清晰地表达出来。”

一系列新的计算机程序在此基础上相继问世，引起了巨大的轰动。一款叫做“专家系统”（expert system）的新程序走进了人们的视野。它能够有效地运用某个领域的专家多年积累的有效经验和专门知识，通过模拟专家的思维过程，解决一系列问题。它的用途非常广泛，可用于诊断疾病，也可用于推荐合适的理财产品，这取决于每个专家系统的专长是什么。研究人员渐渐认识到，他们可以用计算机来处理自然语言，即自动对句子结构进行划分和处理。“聊天机器人”（Chatterbot）是一款用于模拟人类对话或聊天的程序，“Eliza”和“Parry”都是早期非常著名的聊天机器人，这些程序整合了大量语言学资料，

使计算机可以对用户的输入语言进行处理，然后做出相对较为可信的回复。有时候，它们的表现着实令人称奇，短时间内甚至会让人误以为它们是真人。

1963年，麦卡锡来到斯坦福大学，创立了自己的人工智能实验室。同一年，明斯基和西摩尔·派普特（Seymour Papert）在麻省理工学院联合创立了人工智能研究小组（Artificial Intelligence Group，即现在的麻省理工学人工智能实验室），以开展数学与计算项目（Project on Mathematics and Computation）。人工智能领域由此形成两派分庭抗礼之势。美国西海岸的研究人员以麦卡锡为首，称为“简约派”（the Neats），因为他们注重逻辑和证明。东海岸的研究人员则受明斯基的影响，称为“芜杂派”（the Scruffies）。

明斯基的方法并不是建立在数理逻辑的基础之上。他认为，单靠逻辑不足以创造出人工智能。所以他另辟蹊径，引入了一个新的概念，称为“框架”（frame）。我们知道，现实世界中的信息并不总是符合逻辑的，但它们一定可以反映这个世界的真实情况。框架^{[\(10\)](#)}就是一种可以存储这些“芜杂”数据的数据结构。根据语义网络理论，框架里的信息由多个槽^{[\(11\)](#)}、值^{[\(12\)](#)}以及类型组成。下面是一个描述“大象”的框架：

槽	值	类型
大象	-	(子框架)
类属	哺乳动物	(母框架)
足数	缺省 ¹⁶⁵ : 4	(缺省,由哺乳动物框架继承而来)
性别	雄性或雌性	(过程附件 ¹⁶⁶)
耳朵大小	大	(实例值)

165 缺省，即系统默认状态，意思与“默认”相同。

166 过程附件,表示动态过程的条目,目的在于使框架兼有陈述性和过程性表示功能。过程附件只在特定条件成立时才被触发而起作用，因而称为触发程序或“守护神”。

由于不同的框架之间可以相互联系，并相互分享和继承信息，因此我们可以设计出一款程序，使之能够采用看似智能的方法进行搜索，并根据已有的信息进行推理。明斯基的另一个创举是提出了“微世界”(micro-world)的思想。微世界就是一种高度简化的小型环境，或者说是场景，研究人员可以把它当成人工智能方法的测试平台。举个例子，假设有一个“积木世界”(blocks world)，在这里除了积木就没有别的东西，你要做的事情就是按照确定的顺序，把积木一个一个地搭建起来。这种简化使研究人员可以对多种新型人工智能方法进行开发和测试。

借用上面积木世界的例子，假如我们开发了一种可以看懂自然语言的程序，想要对它进行测试。我们只需对其发出指令，然后观察积木的搭建方式是否符合指令，就知道程序对自然语言的理解程度到底是什么水平了。

“简约派”和“芜杂派”在各自的战线双双告捷，胜利的喜悦让先驱者们信心爆棚，他们对人工智能的未来做出了乐观而又大胆的预测。1965年，赫伯特·西蒙宣称：“不到二十年，机器就能够做到人可以做的任何事情。”1970年，明斯基声称：“不出一代人的时间……创造人工智能的问题就会基本得到解决。”

麦卡锡离开麻省理工学院后的几年里，明斯基和派普特研究了一大批与人工智能有关的问题。在为数众多的问题中，人工神经网络逐渐吸引了他们的注意力。1957年，明斯基的老同学弗兰克·罗森布拉特（Frank Rosenblatt）设计了一个更为先进的神经元模型，称为“感知机”（perceptron）。这个模型与麦卡洛克和皮茨设计的简单神经元模型基本相同，只是对输入信号的权重进行了些许调整，权重值可以在-1到1之间浮动，此外它的阈值也是可以调整的。罗森布拉特还设计了一套拓扑结构图，用于描述神经元之间的联系：神经网络分为三层，它们分别是输入层（input layer）、内部隐藏层（internal

hidden layer）和输出层（output layer）。这个模型的创立引领了20世纪60年代对人工神经网络的研究风潮。不过明斯基由于之前做过人工神经网络，所以并不认同新的模型。明斯基和罗森布拉特之间发生了激烈的争论，这促使明斯基决定把他所有的反对意见都写下来，仔细编撰成一本书。这就是大名鼎鼎的《感知起：计算几何学导论》（Perceptrons: An Introduction to Computational Geometry），该书于1969年出版，给当时火爆的神经网络研究带来了巨大的冲击。

“截至1969年，人们发表了几千篇关于感知机的论文，”明斯基说，“但是我们的书一下子就把讨论感知机的氛围给吹散了。现在回想起来，我觉得这本书对感知机批判的有点过火了。不过我书里的观点归根到底就是一句话——感知机无法将视觉上分散的东西汇总到一起。”说着，明斯基拿起一把勺子，将勺子藏到了碗的后面，“你其实知道这是一把勺子，虽然你并没有看到它的全部，只是看到了它的柄和勺尖。但是感知机做不到，它不能把这些信息汇总起来，得出一个真实可靠的结论。”

约翰·沙维-泰勒教授一直致力于研究人工神经网络，试图找到能让计算机学会学习的方法，也发表过一些价值很高的学术作品。“明斯基的书写得非常好，无论是文笔还是内容都非常出

色，里面列举了大量很有意思的实验结果，让你不得不相信他鼓吹的那套理论。当然，他写的东西也没有错，书里也没有什么虚假的实验结果……问题在于，他的表述方式根本没有给其他观点留活路。而我认为这是非常有害的。”

尽管明斯基后来也承认，他对感知机的简约设计和强大功能还是很钦佩的，但是神经网络研究已然遭到沉重的打击，覆水难收。20世纪70年代初，在明斯基的书出版三年后，人工智能领域迎来了第一个冬天。除了感知机的失败以外，早期自然语言处理项目也遭遇滑铁卢，机器翻译尝试也在讥刺和嘲笑中陷入困境。1973年，英国科学研究委员会（the British Science Research Council）发布了一份报告，对人工智能领域的研究工作进行了深入批判。报告称，人工智能研究耗费了大量的人力物力，其结果却令人失望，与明斯基、西蒙等科学家预计的结果相差十万八千里。除了艾塞克斯大学（University of Essex）、萨塞克斯大学（University of Sussex）和爱丁堡大学（University of Edinburgh）以外，全英国的人工智能研究工作都失去了经济支持。世界各地的人工智能研究也都遇到了类似的经费危机，人才流失严重。

尽管举步维艰，人工智能领域的研究依然在继续开展，不少方法和成果在现实世界相继得到

应用。到了20世纪80年代，小到洗衣机，大到电梯，各式各样的设备都采用了模糊逻辑⁽¹³⁾控制器，操作起来越发地顺畅和智能化。专家系统已经得到广泛应用。新的大型并行计算项目也开始启动，它们的核心正是应用了像LISP这种基于逻辑的计算机语言。一些大型数据项目开始启动，CYCq就是其中之一，这个项目雄心勃勃，旨在建立一个囊括所有常识性知识的巨大知识库。在神经网络领域，新的神经元模型和机器学习方法相继问世，比如霍普菲尔德网络（Hopfield network），和反向传播算法⁽¹⁴⁾。它们的出现似乎提供了强有力的武器，可以帮助人们解决一切之前未能解决的问题。投资界对人工智能的投资触底反弹，人工智能的春天回来了。

但是公众又一次失去了对人工智能的信心。人们发现，建立专家系统需要耗费漫长的时间收集专业知识，而且，如果专家系统遭遇了专业领域以外的任何问题，它们给出的答案常常令人啼笑皆非。几个大型的人工智能计算项目都失败了。很多建立在这些技术基础之上的公司纷纷倒闭，研究经费很快撤离。甚至连最新的神经网络方法也都伴随着一大堆问题——规模大的问题解决不了，规模小的问题又无法保证拿到最优解，速度缓慢而灵活性差，令人失望。

大部分人工智能理论都是建立在符号处理的

基础之上，现在，这一基础也饱受质疑。在符号处理的问题上，简约派和芜杂派区别不大，都在使用计算机操控符号。哲学家约翰·希尔勒⁽¹⁵⁾

（John Searle）将计算机比作一个待在“中文屋子”里的人。屋里的人不懂中文，但我们给这个人传递一条用中文写成的信息，他在文件柜里翻找合适的回复内容，最后用中文给我们回复了一条新的信息，而且由于他不懂中文，自然也不会理解信息的意思。在屋外的人看来，屋里看起来像是有一个以中文为母语的人，但事实上，这个人根本不懂中文。希尔勒认为，只会执行符号处理的计算机永远不会拥有智能，无论它们做出的成果有多么惊人，都不会改变这个事实。即使“聊天机器人”通过了图灵测试，它还是不懂它使用的文字，只不过是一台通晓测试游戏规则、且善于投人所好的邮件分拣机罢了。

人工智能又一次名声扫地，并在20世纪80年代末迎来了第二个冬天。少数坚守阵地的科学家甚至有点害怕使用“人工智能”这个词，转而采用了“智能系统”、“计算智能”、“机器学习”等词语替代。他们默默开展着研究，没有人敢做任何大胆的预测。但是，与此同时，人工智能领域正在悄然成熟，并获得新的生命。

自下而上的人工智能

20世纪90年代初，受哲学家和机器人学工程师的影响，研究人员对人工智能的概念理解产生了动摇。麻省理工学院教授罗德尼·布鲁克斯（Rodney Brooks）抓住了思维转变的要点，撰写了题为《大象不会下棋》的论文，标题恰到好处地点出了人工智能传统认识的局限性。他认为，与任何机器人或是计算机程序相比，大象这类哺乳类动物都要聪明得多。它能在自己的世界里走动，与其他大象交谈，甚至会哀悼死去的大象。但是大象并不会下棋。它们没有逻辑思维，不会推理或是操纵符号。

现代的人工智能程序也许能够做到逻辑推理，也有可能拥有高超的棋艺。但是在真实世界里，它们连一个机器人都控制不了。它们不能处理现实世界中的变化性，多样性以及不协调性。现实世界其实是一团乱麻，传统的人工智能根本无法应付这样的环境。

布鲁克斯认为，传统的人工智能把主次给颠倒了。研究人员应当更多地采用一种以行为学为基础的方法，而不是盲目对大脑进行建模，而后将其输入计算机。智能应该自下而上地构建，而不是相反。“我意识到自己不得不把整个研究思路倒转过来，”布鲁克斯说，“主流的研究思路是自上而下的，他们把大脑的每一块有感知能力的部分都纳入一个统一的模型当中，设计者再根据

这个模型制定计划，每一个任务执行单元都要严格按照计划的程序一步一步执行命令，这太盲目了。生物系统里真实的情况不是这样的，从感知到行动的反应非常快。”

布鲁克斯运用自己对生物学本质的深刻认识，建立了适用于机器人的“包容结构”（subsumption architecture）。他从一层层的本能行为开始构建智能，有些层次的本能行为比其他层次的优先级要高。所以，低层次的行为也是最重要的行为，譬如我们不会随便撞到障碍物，这就是一种本能。高层次的行为重要性相对低一些，像随机性的四处走动就属于这种类型。而探索周围环境可能就属于更高层次的行为了。所有行为都会直接受到传感器的影响，但是低层次的行为会优先处理，这就意味着它们可以随时中断高层次行为的执行。“我们可以在传感器和执行器之间建立速度非常快的联系网络，网络里不是几个传感器和执行器的互联，而是有许许多多多的器件都参与了这个网络的构成。我的思路就是从这里冒出来的，”布鲁克斯说，“我大略地给你讲一下这个进化模型，模型开始的时候可能相对比较简单，随着时间流逝，层次越来越多，模型也越来越复杂，新叠加在上面的部分可能会干扰到较低层次的运作。所以我就想到，我们可以只用非常小的计算量，就能实现能在现实世界运作

的系统。”

采用布鲁克斯的方法后，科学家做出的人工智能机器更加贴近自然状态，这种人工智能机器可以完成一系列复杂的行为，它会根据所处的环境，动态地开关或是组合预设的基本行为。重要的是，它既没有处理任何符号语言，也没有预装的知识库。对布鲁克斯来说，世界本身就是它自己的模型，机器人的行为只是去适应这个世界而已。他们按照这个方式做出的机器人，运行结果令人惊叹。“那个时候，全世界还只有几台会走路的机器人，它们都假设环境是静态的。它们对环境有一定的感知能力，然后计算个十五分钟，走上几英尺，又停下来重新评估周围环境，再计算十五分钟，就这样循环下去。而且全部运算还得在大型计算机上完成，”布鲁克斯说。“但我们的系统不一样，我们的系统里只有一个小型处理器，就装在机器人身上。机器人从造出来的那天起，就能走来走去，而且没有撞到任何东西，如果有人走向它，它会主动避开人。它感受到的环境是动态的，反应也非常快。我们后来毫无保留的把这项技术公开了，没有收人一分钱，这个系统运作异常出色，比传统系统好出一大截。”

菲尔·赫斯本兹是萨塞克斯大学的教授，也是人工智能和机器人学领域的专家。“按照布鲁克

斯的口号，世界就是它自己最好的模型。这似乎不符合常理，如果将世界看成是一个层次，而又没有人能超脱这个世界，那这种模型与和稀泥又有什么区别呢？”他说，“但是这种表述其实非常好，它抓住了一个我们常常忽略的事实。在现实世界里，很多适应性的行为不需要行为代理人

（behaving agent）对外部世界的内部运作细节进行建模。传统的人工智能观点包含了一个中心假设，那就是，所有的行为都是通过对环境建模，然后依据模型进行逻辑推理后才产生的。布鲁克斯证明，建模并不是产生行为的必要条件，事实上，利用他的方法，我们可以让智能机器完成许多令人印象深刻的实时行为，有些行为是当时的传统人工智能机器无法实现的。”

这个思路并不都是全新的。罗斯·阿什比

（Ross Ashby）是英国的一位人工智能先驱，他和图灵都是比例俱乐部的成员。1948年，他发明的一台仪器使用了与布鲁克斯类似的原则。同态调节器（homeostat）是一种特殊的机器，它甚至在受到外界扰动的情況下，也可以维持自己内部的稳定行为。它可能看起来就像是桌子上的一组盒子，但是笨拙的外表下，这个小机器可以自主调节自己的行为，令人颇感意外。以至于阿什比在时代周刊的一篇文章中声称，这台机器是“人类迄今为止所制造的机器中最接近人工大脑

的。”差不多在同一时间，比例俱乐部的另一名成员格雷·沃尔特（Grey Walter）向人们展示了他制作的机器人“乌龟”，这个机器人可以自主移动，它身上携带有简单的传感器和马达，可以完成相当复杂的行为。沃尔特的工作对于布鲁克斯等自主性机器人学的先驱来说，是一个重大的启迪。沃尔特在描述自己的机器人时写道：“这个模型看起来可以不像动物，但是它的行为一定要像。因此，它必须具有探索心、好奇心和自由意志，至少三者必须有一。自由意志表现在很多方面，比如不可预知性、目标搜寻、自我调节、趋利避害、前瞻性、记忆力、学习能力、遗忘性、联想能力、形状识别，还有就是一些能使它适应社会的要素。这才叫生命。”

布鲁克斯的思路清晰而又明确。在20世纪90年代初，他的观点已经日渐为人们所接收，这也标志着人工智能领域的研究进入了一个全新的阶段。一夜之间，研究人员就可以尽情使用非符号性方法探索智能的奥秘了。差不多在同一时间，科学家斯图尔特·威尔逊（Stewart Wilson）发表了一篇论文，题为《人工动物：实现人工智能的必由之路》（The Animat Path to AI）。他在文中第一次提出了“人工动物”（animat）的概念（这个术语既可以指机器人，也可以指虚拟仿真技术），开拓了一片新的研究领域，引起了学术界

的广泛关注。他的观点受图灵影响很深，他认为，研究人员应该在逐渐复杂的环境下，开展人工动物的仿真工作。与布鲁克斯类似，他也相信智能应该是自下而上构建的。“人工动物理论的基本策略是，采用最少的特制机械，从底层做起，逐步向更高层次的智能迈进。”威尔逊说。他建议，研究人员应该先制造可以在简单环境下运行的简单人工生命，然后缓慢增加环境的复杂性，或是在同样的环境中，增加人工生命所需完成任务的复杂性。而且人工动物的复杂性不可以随意增加，只要这个复杂性足以使人工动物适应新环境就够了。在理想条件下，应该尽可能地使用自适应式的解决方案。用这种方法的话，一旦有需要，人工动物的智力就能自己发挥作用，就像生命在现实的自然环境中进化一样。这可以让研究人员了解大脑中的每个要素是如何出现、并且和其他要素一起发挥作用的。凡是与生命的感知、运动、预测和生存相关联的问题都应引起人们的重视，只要对它们进行深入研究，人们就能更加透彻地理解更高层次问题，比如交流、计划还有逻辑推理。自上而下的研究思路已经在人工智能领域占据了几十年的统治地位，布鲁克斯的思路则将其彻底颠覆。人工智能领域的气象从此焕然一新。

没过多久，科学家们就召开了新的大型计算

机科学会议，专门讨论自适应行为的仿真和人工生命。研究人员已经不再只是跟着数学家亦步亦趋，拾人牙慧式地探究智力的奥秘。如今，计算机科学家正在与生物学家和哲学家紧密合作，共同探索智力和生命本身的起源。

适应：自然选择的产物

大屏幕上播放着一段视频。视频中，有一个不明物体在游动，从它划水的动作看，感觉像是一只乌龟，它的身体由一系列小方块构成。紧接着，另一个奇怪的生物进入到画面当中。它的块状身体上有三对连续排列的桨。“这个演示视频展示的是，在模拟的物理环境中，为了完成特定任务而进化出来的虚拟生物。”画面中传来了一句美式口音的旁白。

话音刚落，又一种生物进入了观众的视野，它像蛇和鳗鱼一样，以波浪式的动作向前游动。“我们用游动的速度作为判断它们能否生存下来的标准，”旁白接着说，“大部分的虚拟生物都是独立进化的产物，有些还发展出了与真实生物相似的生存策略。”

屏幕上出现了好几条泳姿优美的蛇形生物，它们都由链状方块构成；还有三只像蝌蚪一样的虚拟生物以优雅的姿势游弋于水底，时不时掠过河床。“一旦生物发生了进化，它们的拷贝数就

会增加。在相同的环境中，它们表现出来的特性也基本相同。”

视频的画面切换到了陆地上，一只长得像虫的生物不断地扭动身体向前运动。“这种生物先是在水里进化出了游泳技能，然后我们把它放到陆地上，它又一次进化了，成功地获得了侧向运动的能力。”

这时候，画面中出现了一只形似海龟的虚拟动物，看起来，它在水中的灵活性很强，四条短腿像桨一般划动。它被一颗红色的光斑吸引，改变了方向，跟着这颗光斑游动起来。“这组虚拟生物进化出了跟踪红色光源的能力，”旁白以一贯冷静的口吻说道，“我们可以看到，它们正在与光源互动。随着光源位置的改变，它们的运动状态也会发生相应的变化。”

视频里介绍还介绍了很多新的生物，此处暂且按下不表。这段视频于1994年首次对外发布，名为《虚拟生物》（Virtual Creatures），制作者是卡尔·西姆斯（Karl Sims）。如果说，实现人工智能的新方法是进化式地缓慢增加程序的复杂性，那么，《虚拟生物》可能就是这种方法运用于实际的巅峰之作了。同样值得一提的是，即使是在与进化相关的计算机科学领域中，《虚拟生物》也可以算是留名青史的重要成果，它代表了进化计算以及自适应行为仿真领域的早期成就，

显示出计算机在智能进化研究上的巨大潜力。西姆斯给他的虚拟生物制定的生存规则并不复杂，只是要求它们游得更好、走得更好而已。配合逼真的物理和进化模型，虚拟生物的进化历程给人以强烈的似曾相识之感，它们与我们平日里熟知的那些真实生物在形态和动作上都惊人的相似。不过，在进化计算领域首开先河的并不是卡尔·西姆斯，而是另有其人。早在计算机科学诞生的时候，第一批获得计算机科学博士学位的研究者就有人做出了前瞻性的成果，奠定了进化计算的基础。

约翰·霍兰德1929年出生于美国印第安纳州的韦恩堡。他在邻州俄亥俄的一个小镇长大（按照他的发音习惯，“俄亥俄”会被念成“俄亥阿”）。霍兰德是个聪明的孩子，父母从小就给他买来化学实验玩具任其摆弄，母亲也会和他对弈西洋棋。霍兰德还对制造玩具潜水艇等小机械抱有浓厚的兴趣。与明斯基等计算机先驱者不同，霍兰德一直没有机会进入培养特长的专门学校，只能在当地一家规模很小的学校读书。学校里也没有老师能够对他因材施教。随着霍兰德日渐长大，等他准备报考大学的时候，问题就暴露了。“高中的时候，我得找一个做给排水工程的人来教我三角学。学校里没有这门课程，但如果我想去麻省理工学院读书的话，三角学就是必修课。”霍

兰德说。他从小就对数学很感兴趣。事实上，正是由于生物学缺乏数学基础，他才对这个学科丧失了兴趣。“高中的时候，我对生物非常的感兴趣。但是我不知道怎样才能生物学里使用数学。如果生物学里能明确运用较为严格的数学语言，那我可能就会去研究生物学了。”

虽然没有受过优越的精英教育，霍兰德依然在全国物理联考中斩获了第三名，他也因为这次出色的表现获得了麻省理工学院的奖学金。1946年，霍兰德入读麻省理工学院，主攻物理。大四那年，他在新问世计算机“旋风”上完成自己的学士学位论文。这段宝贵的经历使他加入了IBM的第一台商用计算机701的规划小组。霍兰德很快就发现，是组里唯一一个有过实际计算机编程经验的人。“实验室里组装机器的同事们都没有编程经验，我给他们上了一堂简单的编程课。”他回忆道。

当时，有好几名学者正在访问IBM，这让霍兰德有机会见到了麦卡锡、约瑟夫·利克莱德、约翰·冯·诺依曼等计算机先驱。麦卡锡还教他学会了下围棋。在IBM工作期间，霍兰德与上司罗切斯特一起开发了神经网络模型，成功发明了第一个赫布型神经网络。在研发的过程中，他们数次前往在麦吉尔大学（McGill University），拜会了唐纳德·赫布。十八个月后，霍兰德决定攻读读博

士。IBM允许他夏天上班，冬天上学。由于密歇根大学的数学系非常强，他选择了入读密歇根大学。

读博期间，霍兰德一门心思扑在了数学上。当他遇到阿瑟·伯克斯（Arthur Burks）的时候，他甚至已经开始撰写关于圆柱代数（布尔代数的衍生分支）的论文了。伯克斯那时候正在筹建一个通信科学项目，他认为霍兰德可能会感兴趣，想拉着他一起干。伯克斯对计算机的理解非常深刻，大部分人懂的都没他多，他是世界上第一台电子计算机“埃尼阿克”项目的高级工程师，作为咨询顾问，参与了离散变量自动计算机“埃德瓦克”的设计，他是埃克特的上司，莫奇利的室友，后来他又在普林斯顿高等研究院与冯·诺依曼一起工作。他在摩尔学院还讲过九次课。

霍兰德觉得要加强自己在心理学和语言学方面的知识功底，最后，他完成的博士课题是伯克斯的抽象逻辑网络，这个网络与麦卡洛克和皮茨的神经网络非常相似。1959年，霍兰德刚毕业那阵子，正好赶上密歇根大学成立通信与计算机科学系。“我是世界上最早拿到计算机科学博士学位的人，那时还是在1959年。”霍兰德说。

毕业后，霍兰德决定留校任教（后来他在密歇根大学一直工作到了退休）。没多久，业界开始流行一个新的研究思路：让虚拟生物在自然选

择的压力下适应环境。霍兰德读了英国数理统计学家罗纳德·费希尔（Ronald Fisher）的著作《自然选择的遗传原理》（The Genetical Theory of Natural Selection）。这本数学著作介绍了自然进化研究上的一些重要概念，例如生物体进化时，适应度会根据所处环境的不同而发生改变，这个过程似乎与其基因的变化有对应关系。霍兰德被书中的数学所吸引，但他很快就发现有些内容还不完善，需要进一步的解释。费舍尔似乎暗示人们，基因之间是相互独立的。霍兰德知道实际情况并非如此。他通过大量的阅读了解到，一组基因可以协同发生作用，一起为生物体创造有利的元素。这一组组的基因就像积木一样，可以反复利用，从胡萝卜到大象，每种生物都是按照这个原则进化的。“我认为，即便是等位基因簇，也不是独立工作的，它们之间也存在相互作用，”霍兰德表示，“费希尔假设等位基因的效果是叠加的，而我认为等位基因簇之间存在相互作用。对我来说，实现自己的假设也是个不小的挑战。”

霍兰德的解决办法是创建一个计算机模型，他称之为遗传算法。这个算法是一个进化的模型，它带有一个由很多独立个体构成的种群，其中每个个体都包含了一组基因。基因用二进制数码表示。为了模拟自然选择，霍兰德首先做出了

一个适应度函数，将需要解决的问题抽象画，然后后后用这个函数去评价每组二进制数码（等位基因）的影响。如果数码组能够解决问题，就会获得较高的适应度评分。适应度评分较高的个体就被作为亲代，用来培育新的后代种群。后代会从父方继承一些基因，也会从母方继承一些基因，还可能伴随着一些随机突变。然后再用适应度函数给新一代的个体打分，较好的个体又被作为亲代。这样一代代地进行下去，就等于在计算机程序中实现了进化。于是霍兰德就可以在虚拟生物进化的过程中，观察和分析等位基因簇的变化频率了。

“共适应的多组基因是个很重要的概念，所以我希望在它们身上重复费舍尔在等位基因上做过的工作。我问自己，如果我试着把它们当成超级等位基因，会出现什么情况。在数学上我能做些什么？”霍兰德提出了他的“积木块假设”（building block hypothesis），这个假设认为，为了获得优秀的解决方案，进化并不是一次只改变一个基因。进化会去寻找优秀的等位基因簇，将它们保存下来，像搭积木一样，让生物体将它们继承下去。直到现在，霍兰德依然相信，这个假设是普遍的原则性概念，我们无时无刻都需要它。“我认为，如果将这个概念推而广之，我们可以看到，各种各样的事情中都存在这个现

象。标准积木的重新组合是非常普遍的。我就拿内燃机举例好了，内燃机上的几乎每个零件都不是什么陌生玩意，都是人们已经用了一百年的东西，但它们的独特组合才是最重要的，单个部件其实并没有我们原来想象的那般重要。”

霍兰德在20世纪60年代完善了自己的思想。他引入了“分类器系统”（classifier system）的概念，这是一个介于专家系统和遗传算法之间的理论，分类器系统可以自己学习规则，也就是说，不需要给它确定规则，它就可以自适应的调整搜索方向。1975年，霍兰德将这一概念写入书中发表，书名为《自然系统和人工系统中的适应》

（Adaptation in Natural and Artificial Systems）。可能由于传统人工智能观点的统治力太强，霍兰德的遗传算法一直未受到充分重视，直到1989年，他的学生戴维·戈德堡（David Goldberg）出版了有关遗传算法的新作，进化计算这个领域才开始真正腾飞。没过多久，进化算法就被用于最优化^{[\[16\]](#)}，目的是为了发现新的数学函数，或者为机器学习软件中的自适应打基础。可以说，整个计算机科学界刮起了一阵进化算法的旋风。事实证明，进化算法特别擅长处理那些别的方法解决不了的麻烦问题。

其他偷师于大自然的算法也被接二连三地开发出来。有些参考了蚂蚁种群里的蚁群运动模

式。霍兰德的另一个学生斯特凡妮·福里斯特（Stephanie Forrest）根据免疫系统的运作机理发明了一种新的算法。还有一个学生克里斯·兰顿（Chris Langton）成为了人工生命领域的奠基人之一。进化计算领域的最新研究着眼于一些基础性问题。比如胚胎的形成、生物的发育过程等生长过程怎样才能用进化算法实现？在此基础上，怎样才能开发出更加复杂的算法？如今，进化学习已经成为人工生命和机器人学领域的重要组成部分。很多公司利用它的成果，开发出了多种多样的应用性产品，手机的天线设计还有存储芯片就是两个典型案例。

直到今天，霍兰德仍然在进行进化和学习方面的研究。他的理念一向基于自下而上的学习体系，甚至是当年这个观点还没有被广泛接受的时候，他就已经对其深信不疑。“我非常认同罗德·布鲁克斯的观点。首先要做的就是找准定位，”霍兰德说，“必须存在一种环境，使人工智能体可以与它互动，换句话说就是，人工智能体要在环境的压力下生存下来。我从一开始就是这样想的。我不相信机器可以不通过学习和适应就获得真正的智能。”

学习如何学习，预测如何预测

近些年来，人工智能教会了我们很多东西。

譬如，我们已经认识到智能非常复杂，这恐怕是最重要的经验了。自上而下的逻辑编程也好，自下而上的进化适应方法也好，无论采用哪种方案，都无法解决所有的问题。没有任何一款程序能够通过图灵测试。我们还是很难制造出一台具备智能的机器，无法在不可预测的环境中操控机器人。事实上，现在的双足机器人离人还太原始，即使想让它以自然的方式行走都极其困难。

但是，只要把人工智能的问题化整为零，各个击破，我们就能取得一定的进步。近年来，人工智能领域取得的最重要的突破之一，表现在机器学习领域，也就是制造可以学习的计算机程序。我们或许还不能像教小孩子一样教计算机学习知识。但是，在机器学习领域，我们可以教会计算机如何区别不同数据集合（这个问题被称为分类），或者学习一个数据流的结构，然后对下面可能出现的情况做出预测。如果机器能够区别不同的数据类型，那它们就拥有了决策能力：图上画的是小汽车还是公共汽车？从这个基因序列来看，哪种遗传疾病的发病率更高？如果机器可以对这些问题作出预测，那它们或许就可以预测更复杂的问题。比如，某种股票交易策略能否赚钱；某种药物能否达到预期效果；某种网络数据流模式是否正常，是否有被黑客入侵的可能。

进化计算为机器学习提供了一种著名算法，

那就是遗传编程。遗传编程会像基因一样进化，这主要体现在数学函数和遗传算子上，而非由基因原始序列转化而来的二进制数据串。只要给定一组输入数据和一些正确的样本输出数据，遗传编程就会先对一组候选函数进行比较，然后进化出一套程序或数学函数，以根据输入数据得出正确的结果。这个方法的好处在于，我们可以对进化后的方程或模型进行详细研究。比尔·兰登是顶尖的遗传编程专家。“基因编程的优势在于，这种方法可以帮助我们做出易于理解的模型，”他说，“以前我在葛兰素史克（Glaxo Smith Kline，简称GSK）工作的时候，经常对药物建模。有一次我和同事参加了一个为期两天的小型研讨会，最后一天下午，我们上台做报告。当时研讨会里来了很多机器学习方面的专家，他们也讲了不少。我的报告里有一张幻灯，上面提到了遗传编程模型。别人都没有提到过这个问题。会议厅里的观众一下子就炸了锅，我听到有人喊：‘约翰，你看吧，我两年前就跟你说这玩意重要，你他妈就是不听！’关键在于，他们不仅理解这个模型里的内容，也很喜欢它。对他们来说，这个模型的智能化很强，而且每个构成部分都是他们所熟悉的。”

神经网络为计算机学习提供了另一个重要的方法。感知机和反向传播算法（该算法可以自动

调整神经元输入信号的权重)的局限性曾经引发过学术界的猛烈批判,但它们对学术界的影响还是非常明显的,后来在它们的基础上,研究人员又开发出了很多新型神经网络,调整权重和连通性的巧妙办法也层出不穷。这些研究性质的神经网络之所以被发明出来,通常是出于两个目的。一是为了提高机器的人工智能,二是也是为了进一步的了解大脑运作的内在机理。吉奥夫·辛顿(Geoff Hinton)是最早将反向传播算法应用于神经网络的科学家,他还发明了另一种算法,称为“玻尔兹曼机”(Boltzman Machine)。他对这些领域的认识清醒而又深刻。“我对思维方式和大脑的计算模式很感兴趣,”他说,“在我看来,这两个问题基本一致。从学生时代开始,我就想把它们好好研究一下了。”

机器学习的最新突破与生物学联系较少,反而与数学联系较多。一种新的基于统计学思想的机器学习方法登上了历史的舞台,那就是支持向量机(Support Vector Machine,简称SVM)。尽管它只不过类似于一个简单的神经元,但研究人员在此基础上做了大量深入的研究工作,建立了一整套机器学理论。约翰·沙维-泰勒教授是该领域的顶尖科学家,他给我们讲解了支持向量机是怎样学习的。假如计算机需要弄清楚报纸上某篇文章的内容,或者帮助用户寻找特定的信

息。“那么，计算机首先需要判断的是这篇文章所属的话题，”沙维-泰勒说，“譬如它是不是体育类的。根据文中出现的某些词汇，计算机就可以大体判断出这篇文章是不是属于体育类。如果出现了“足球”这样的词语，那么它属于体育类的几率就会比较大。当然，情况也不能一概而论，也许它讲述的是有关体育政治的话题，但好歹计算机的初步判断错的并不离谱。由此可见，（分类器的）决策建立在对目标特征进行考量的基础之上。这样的分析思路贴近事物的本质，支持向量机就是采取了这样的学习机制。在这个例子中，分析对象是一篇文章，但其实任何东西都可以拿来分析。”

支持向量机会考察情景中特征的加权化组合，然后据此进行决策。简单的感知机神经网络首先将输入信号转化为加权和，然后进行反应。支持向量机和它并没有太大的区别。但是支持向量机的强大之处在于，我们不需要详细阐明所有的特征，只需要定义不同对象之间的“相似性函数”。比方说，如果报纸上登载的一篇文章如果与另一篇共用了许多普通词语，那它们就可能是相似的。支持向量机会调用相似性函数，用以确定从文章转化而来的两个数据集之间的距离有多远，即确定哪个数据集所代表的文章属于体育类，而哪一个不是。“从某种意义上说，它也属

于经验式的方法，”沙维-泰勒说。“但这个方法有着很坚实的理论统计分析基础。你只要深入研究就会发现，你对算法的优化和最后的预测结果息息相关。”

如今，科学家已经发明了多种类型的支持向量机，此外还有一些其他的、基于统计学的机器学习方法。有些算法可以使计算机计算不同事物相联系的概率，譬如贝叶斯学习法（Bayesian learning）。这种“讨巧”的方法使计算机学习极其复杂的关系问题成为了可能。“对我来说，这就好比在原来的方法中加了点概率因素，使人工智能的问题就变得好解决了一些，这是个一个带有概率性质的人工智能，它能让你完成一些看起来困难得多的任务，这就是它的优势所在，”沙维-泰勒说。“有的时候把问题化整为零，解决起来就简单了。理论上看起来很复杂的任务，其实用带有概率性质的人工智能来解决会更简单。”

随着科学技术的日新月异，如今的计算机已经可以实现香农、图灵、冯·诺依曼等先驱提出的许多奇思妙想。有了精巧的图像处理软件，计算机可以认读汽车牌照，辨认指纹、虹膜，甚至辨识人脸。它们可以听懂人声，使用户无需打字就可以传达信息。如今，连最便宜的数码相机都可以自动对焦，自动弥补画面中的缺陷，识别拍摄所属的环境类型。计算机会根据用户的网购模式

揣测其偏好，推荐其他的商品。一旦用户的信用卡被盗，计算机就会发现交易中的异常情况，并发出警告。计算机可以控制汽车发动机、洗衣机等成千上万种设备，使它们运转更高效，更好地服务于人的需要。随着机器人真空吸尘器、自动机器人玩具等产品的相继问世，人工智能已经渗透到人类生活的方方面面。或许现有的机器还不够智能，离科学家的预期尚有一段距离，但技术的发展并未止步，人工智能势将取得更大的突破，解决更加复杂的问题。

充满变数的未来

长久以来，人工智能及其相关领域——启发式搜索、人造生命、进化计算、机器学习等问题不断给人类带来挑战。面对这些深层次的基础性问题，我们知道得越多，问题也就越多。在探索的旅途中，人们得以从全新的视角审视了困扰哲学家数百年的问题：人为什么存在？生命的起源是什么？生命体何以进化出了如此复杂多样的形式？智能是什么？意识是什么？

如今，为了回答这些问题，人们广泛采用了代理人基⁽¹⁷⁾模型（agent-based model，简称ABM）。这种新型人工智能对一系列问题建模，建模对象包括个体（即代理人）、个体行为、个体之间的互动、个体与环境的互动等等。代理人

可以是携带基因、不断进化的生物，与进化计算一样；可以是由计算机操控的物理机器人，彼此之间通过光、声音和肢体接触交流；也可以是合成生物学的产物——由生物学家和计算机科学家合作培养的人造细胞培植而成。它们的模型可以是细胞自动机⁽¹⁸⁾，也可以是人工化学的精细模型。它们的行动依据可以是逻辑和推理，可以是从小人身上模仿的情感，也可以是综合已有知识框架和周遭环境做出的分析。所有模型的构建其实都是为了服务于同一个目的——即了解人类自身。“由于代理人可以自主学习和进化，这对我们研究人工智能的基础问题非常关键。”约翰·霍兰德表示。

在新墨西哥州的圣菲研究所（Sante Fe Institute），约翰·霍兰德等众多科学家花费了大量的时间和心血研究复杂系统⁽¹⁹⁾。作为一名研究了几十年复杂系统的科学家，霍兰德理应知道一个的残酷事实，那就是，有很多自然现象和人为现象都是无法用数学工具预测的，只不过要是把每个现象放到显微镜下，化整为零，逐个击破，那么掌握起来似乎并不复杂。“我一直很喜欢这样的理念，那就是，用简单的规则创造复杂的事物，”他说。他要表达的意思很直白。复杂的经济活动是由无数较为简单的交易活动构成的。大千世界中，有数之不尽的动植物在繁衍生

息，它们之间的相互作用无论从数量上看还是从复杂程度上看，都超乎人们的想象。正是由于这种复杂的相互作用，生态系统才得以维持稳定的平衡。细胞和蛋白质的相互作用使免疫系统能够正常运转，为身体的健康保驾护航。虽然生物体的发育过程由基因决定，但这个过程同样离不开基因、蛋白质和细胞的相互作用。人之所以能思考，离不开数十亿神经元之间的相互作用。通过代理人基计算机模型，人们对上述所有过程都有了全新的认识。

长久以来，大脑的奥秘令无数人着迷不已。计算机科学家对于人类如何思考、为什么思考的问题提出了自己的假设。然而，很多深层次的哲学问题依然存在。就算计算机能够模拟大脑的运行机制，那就一定能说明它真的在思考吗？“假如有一台巨型计算机，它不仅能够模拟人脑中的所有电信号，还能够对突触所在环境的化学物质及其浓度变化进行建模，那么这台计算机就应该可以称得上是具备思考能力了，”吉奥夫·辛顿表示，“毕竟，它模拟了真实的大脑活动，既然真实的大脑在思考，那么，为什么不能说模拟的大脑也在思考呢？这就是我的观点。”

怎样才能界定计算机是否在思考呢？如果计算机通过了图灵测试，那么，人类亲手缔造的数字智能或许就能与人类自身相媲美。或许我们可

以假设，不管看起来多么智能的计算机。只要假以时日，就会暴露出它不会思考的本质。“但问题是我们要考察它多久呢？”比尔·兰登说，“你一直研究这台计算机，研究好几年，可是你还是确定不了。而与此同时，它该是什么样还是什么样，也许从学校毕业了，搞不好还在股票市场上大赚了一把，但你还是不信。”

机器人研究先驱欧文·霍兰德教授是少数几个对人工智能问题想得更远的人。“人们之所以制造机器人，肯定是受到了生物学的启发。在我看来，这就和制造工具一样，只不过制造机器人的目的是为了研究生物系统的行为，尤其是人类的行为，我们需要制造出有意识的机器人，这样才能更加深入地研究意识的本质。但是，到目前为止，没有任何证据证明，有意识的机器人比普通机器人、乃至低级机器人（比如僵尸）更好。事实上，我们有理由相信，有意识的机器人可能会变得穷凶极恶，就像《银河系漫游指南》（The Hitchhiker's Guide to the Galaxy）里的马文

（Marvin）一样。但是，我觉得，机器人具有意识只是时间早晚的问题。我们一直在开发基于模型的预测方法，使机器人具备与周围环境打交道的能力，而且这种方法很可能是与环境打交道最好的方法，毕竟，这些都是人类智慧的结晶。对智能机器人的探索工作很可能反而引火烧身，有

意识的机器人会给我们带来意想不到的副作用。”

很多人对这个想法不以为然。约翰·沙维-泰勒就是其中之一。“也许我们会意识到，其实智慧并不是什么重要的东西，真正重要的是经验。我们有经验，但是机器没有，”他说，“有一位哲学家曾经说过，机器不能像人类那样体验生活，这一点我很赞同。如果你愿意，你完全可以把经验本身看成意识。有人说，也许我们只是自以为有经验而已，一切都是幻觉。我对这种说法不反感，但也不会太买账。（也许这是因为我是台挺差劲的机器吧。）我能理解这种说法的逻辑，但它与我对现实的认识差距太大。”

约翰·霍兰德依然对人工智能的前景保持着乐观的态度。“当我们开始建造有意识的机器时，估计大多数人都会这样想：‘糟糕，这玩意竟然也有意识了。’不过，不管怎么说，它们和人类还是有很大不同的，”他说，“我确实希望人类能获得更多有关意识方面的知识。在我看来，这个目标一旦达成，马上就会给人类的学习和沟通交流带来巨大的影响。我真的相信，只要付出时间和努力，这是可以做到的。”

或许，对于人工智能领域现存的挑战，艾伦·图灵有一句话总结得最为精辟：“虽然前进的道路只有一小段是明晰的，但纵使是这一小段探索

的道路，也充满了艰险的挑战。”

注释

(1) 在数学上，试误法是寻觅数学模式最佳解的方法之一，其作法是设定一些决策问题的可控制变量值，检验其能否满足各限制式，若能满足才推算其目标函数值；若不能满足各限制式，则寻觅另一组可控制变量值测试之；最后挑选最佳目标函数值的可控制变量值即得最佳解。

(2) 控制论（台湾称为模控学）研究动物（如人）与机器之间通讯的规律。

(3) 计算机科学的两大基础目标，就是发现可证明其执行效率良好且可得最佳解或次佳解的算法。而启发式算法则试图一次完成一个或全部目标。例如它常能发现很不错的解，但也没办法证明它不会得到较坏的解；它通常可在合理时间解出答案，但也没办法知道它是否每次都可以这样的速度求解。

(4) 搜索算法是利用计算机的高性能来有目的的穷举一个问题解空间的部分或所有的可能情况，从而求出问题的解的一种方法。

(5) 非数值主要指的是数之外的数据，它的操作主要是对数据进行处理，如代码转换、字符处理、排序、检索等。

(6) 在电子学中，硬件描述语言是用来描述电子电路（特别是数字电路）功能、行为的语言，可以在寄存器传输级、行为级、逻辑门级等对数字电路系统进行描述。硬件描述语言在很多地方可能和传统的软件编程语言类似，但是最大的区别是，前者能够对于

硬件电路的时序特性进行描述。

(7) 这里指的是人工智能领域经历的两次冬天，时间分别是在20世纪70年代和90年代。

(8) 比如，铃声响时一个神经元被激发，在同一时间食物的出现会激发附近的另一个神经元，那么这两个神经元间的联系就会强化，形成一个细胞回路，记住这两个事物之间存在着联系。

(9) 分时是指多个用户分享使用同一台计算机。多个程序分时共享硬件和软件资源。

(10) 框架适合表达结构性的知识。所以，概念、对象等知识最适于用框架表示。其实，框架的槽就是对象的属性或状态，槽值就是属性值或状态值。不仅如此，框架还可以表示行为（动作），所以，有些过程性事件或情节也可用框架网络来表示。

(11) 见本注释（18）。

(12) 见本注释（18）。

(13) 模糊逻辑是处理部分真实概念的布尔逻辑扩展。经典逻辑坚持所有事物（陈述）都可以用二元项（0或1，黑或白，是或否）来表达，而模糊逻辑用真实度替代了布尔真值。这些陈述表示实际上接近于日常人们的问题和语意陈述，因为“真实”和结果在多数时候是部分（非二元）的和 / 或不精确的（不准确的，不清晰的，模糊的）。

(14) 反向传播算法（BP算法）主要由两个环节（激励传播、权重更新）反复循环迭代，直到网络的响应达到预定的目标范围为止。

(15) 约翰·罗杰斯·希尔勒，出生于美国丹佛，是一位在加州大学伯克利分校执教的哲学教授。他对语言哲学、心灵哲学和理智等问题的探讨作出了重要的

贡献。此外他还讨论了社会制造的现实和物理现实的特征以及实践推理。

(16) 最优化是在一定约束之下如何选取某些因素的值使某项指标达到最优的一门学科。可解释为可以用来改进某些数量值的方法。

(17) 有别于统计学和计量经济学重探讨变数之间的因果关係，ABM这个研究方法强调解答过程性的问题（“how” questions 或是 “what if” questions），亦即藉由电脑程序模拟代理人（agents）之间的互动来观察所欲探讨的复杂现象如何形成，并依据观察到的样貌（patterns）来检证理论，或是进一步来开创新理论。

(18) 细胞自动机，又称格状自动机、元胞自动机，是一种离散模型，在可算性理论、数学及理论生物学都有相关研究。它是由无限个有规律、坚硬的方格组成，每格均处于一种有限状态。整个格网可以是任何有限维的。同时也是离散的。每格于 t 时的态由 $t-1$ 时的一集有限格（这集叫那格的邻域）的态决定。每一格的“邻居”都是已被固定的。（一格可以是自己的邻居。）每次演进时，每格均遵从同一规矩一齐演进。

(19) 复杂系统是具有中等数目基于局部信息做出行动的智能性、自适应性主体的系统。

计算机改变了我的生活

当今时代是一个数字化的时代。正因为图灵、香农、冯·诺依曼等计算机先驱的开创性工作，如今的人们才得以掌握了各式各样令人称奇的前沿技术。严丝合缝的数学基础造就了无处不在的硅芯片计算机，也造就了功能强大的应用软件。简洁易用的人机交互界面将世界各地的人紧密联系在一起。智能化的机器正以超乎想象的方式给人类的生活带来福祉。

见证奇迹的旅程才刚刚开始。借助前沿尖端的计算机科技和天马行空的想象力，勇于开拓的计算机先驱每年都在前赴后继地编写出令人惊叹的应用软件。如今，艺术家用计算机创作赏心悦目的作品，生物学家用计算机探索人体的奥秘，医护人员用计算机诊治疑难杂症。市场监管者甚至也用上了计算机，以便更高效地打击犯罪。这是一个振奋人心的时代。

年轻的艺术家走进宽敞的实验室，腋下夹着一卷卷画作。讲台上放置着十二台方方正正的大屏显示器，把艺术家围了起来。很显然，所有的显示器都与楼下的主机相连。由于百叶窗没有打开，屋子里光线很暗，使屏幕上各种稀奇古怪的图像显得格外明亮。

艺术家有些紧张，他打开卷起的画作，将它

们摊放在实验室的地板上，这些画作都是他精心构思后绘制出来的，整个地板的空间都被铺得满满的。其中有一幅画很大，看起来像是一件抽象的印度教艺术作品，上面有一条蛇或是一根触角正从画面中心呈螺旋状向外延伸，如漩涡一般。另一幅画也颇为诡异，看起来像是在描绘一个甲虫展览：无数不同形状的甲虫被随机摆放在一起，相似的甲虫彼此相邻。还有一幅画上描绘了一棵进化树，只不过树上的图形颇为抽象，从维京人的头盔到蜂巢，无奇不有。实验室里的赏画者多半都是科学家，少部分是计算机程序员，大家从来没有见过这么古怪的作品。

此前，艺术家一直在努力争取业界的认可。但是他用计算机创作的方式太过前卫，艺术界暂时还无法接受。这次的观众会有什么不同呢？威廉·莱瑟姆扫了一眼台下的观众，开始讲解起自己的作品来。

计算机的创造力

莱瑟姆绝非普通的艺术家。他出生于1961年，在英格兰的布卢伯里（Blewbury）长大。布卢伯里地处农村，但由于临近哈维尔原子能研究所（Harwell Atomic Energy Research Establishment），莱瑟姆从小就接触了很多科学家。他的父亲是实业界的化学家，母亲是作曲家

兼指挥家。科学和艺术的双重影响都体现在了莱瑟姆的身上。他入读了白金汉郡（Buckingham）的斯多中学（Stowe School），这是一所寄宿制的贵族学校，环境优美的园林式校园由著名设计师兰斯洛特·布朗（Lancelot Brown）一手设计。后来，莱瑟姆考取了牛津大学，研习艺术。他早期的作品以造型艺术为主（所谓造型艺术，就是用一定的物质材料塑造可视的平面或立体的形象、反映客观世界具体事物的一种艺术，比如人物肖像画），后来又很快受到了俄国构成主义和自动化创作思潮的影响。（构成主义是反写实主义中最极端的一派，讲究采用抽象的几何形状创造纯粹理智的画面。）“我第一次见识到，自动化系统也可以用于艺术创作，感觉大开眼界，”莱瑟姆回忆道，“从那以后，我开始旁听牛津大学计算实验室的课程，学习了福传语言。实验室里还有一台本森（Benson）公司制造的大型绘图仪，可以将计算机上的数据以图形的形式输出。”莱瑟姆学会了编程的基本原则，但他明白自己依然需要帮助，“我能看到这个方法的潜力，但同时也意识到，自己永远也无法成为一名出色的程序员。”

1979年，尽管莱瑟姆在牛津大学的物理实验室和计算实验室学习了计算机科学，但他依然觉得，要想用计算机和绘图仪创作出合乎心意的作

品，着实不是一件容易的事情。“我碰到了好多挫折，总是拿不出自己想要的成果，”他回忆道。于是，他放弃了采用计算机绘图的想法，转而使用白塞木创作大型雕塑，构筑形似巨大蛛网的结构。1982年，他前往皇家艺术学院（Royal College of Art）学习美术，但他又一次遇到了瓶颈期。怎样才能创作出自己满意的作品？为了寻找这个问题的答案，他花费了大量的时间，在伦敦自然历史博物馆（Natural History Museum）里寻求灵感。这一时期，他开始绘制自然存在的几何形状，如蛇骨架和鹿角。

莱瑟姆的艺术思想还在不断地完善。他创作了大型进化式图像，图中的形状会按照极简单的规则发生改变。“那时候，我把自己的大脑想象成了一台计算机，然后像改良版的绘图仪一样，将脑中的灵感画出来，”莱瑟姆说，“有意思的是，人类的思想可以将扭曲的空间和变形的表面概念化——这些都是机器很难做到的。”由于具备实战经验，莱瑟姆很清楚当时的计算机可以做什么、不可以做什么。1983年，他开始在BBC微型机（BBC Micro）和麦金塔计算机上写程序——不是创作艺术，而是列举一些他绘画时会遵循的指导性意见。系统会从这个规则集中选择一部分，然后列出该画什么。“在这里不存在美学上的边界，”莱瑟姆说。“假如你是个艺术家，你

就要进行创作，总有一些条条框框你必须遵守，说得夸张点就是不敢越雷池一步。所以我的想法就是可以用重复的方法，嗯，就是这个词，不限制重复的次数，很快你就发现，你进入了一个未知的视觉领域。”用这种方法创作出来的绘画作品令人着迷。它们看起来像是非主流艺术

（Outsider Art），一般来说，画出这种作品的人都是些初出茅庐，或是自学成才的艺术家。他们游离于主流艺术界之外，从未被艺术当权派接纳，很有些草莽英雄的味道。

由于皇家艺术学院没有计算机，莱瑟姆便开始借用密德萨斯大学（Middlesex University）和伦敦城市理工学院（City of London Polytechnics）的计算机，并与这些学校的程序员展开合作。没过多久，他听说IBM在温彻斯特有个研究实验室，不同领域的研究人员都可以借用该公司的新型计算机，开展合作研究。于是，他给实验室打电话，介绍了自己的工作。很快，实验室就邀请莱瑟姆过去面谈。温彻斯特之行令莱瑟姆大开眼界，那里不仅有计算机科学家和程序员，还有分子化学家，考古学家，甚至还有外科医生——这名外科医生打算为跳动的心脏建模。对于一名艺海探路的年轻人来说，这里绝对是一块风水宝地。

莱瑟姆把他的绘画作品带到了温彻斯特，做

了个小规模的演讲，介绍了自己用计算机创作艺术作品的想法。出乎预料的是，这个小演讲颇受科学家的欢迎。“他们都觉得这个思路很有意思，”莱瑟姆说，“他们说，‘听起来你想法不少啊。干嘛不过来和我们一起做呢？’”这样的好机会莱瑟姆没法拒绝。“那是我人生的一个重大时刻，”他回忆道。“从那时起，计算机就成为了我生命中非常重要的一部分。”

三个月后，莱瑟姆成为了IBM研究中心的研究员。凑巧的是，此时的计算机图形学领域已经足够发达，能够创制出如照片般逼真、甚至非常精美的图像。数学家史蒂芬·陶德也是计算机图形学方面的专家，他在写一个叫做“可扩展实体模型编辑器”（Extensible Solid Model Editor，简称ESME）的建模程序。这个程序可以用来构建三维模型，但是编辑和修改形状就不那么方便了。他与莱瑟姆合作，按照莱瑟姆设定的艺术规则，再加上用过的随机突变技术，开发了一款新软件。这款软件能够从旧作品中演化出新的作品。两人给这套软件起名为“形状生长与突变子”（Form Grow and Mutator），听起来就像是一个简单的遗传算法。莱瑟姆制定了每个形状的重组和修饰规则，然后将其嵌入程序中。在这里，他扮演了育种员的角色，进化种群中的个体很多，莱瑟姆会决定哪些个体符合系统设定的审美

标准，可以接着产生后代。经过三到四个小时进化，莱瑟姆的突变程序就生成了数百件令人惊叹的全新艺术作品。这些作品都是三维形态的虚拟雕塑，你可以从各种角度观察雕塑的纹理和图案，光线的明暗也可以随意调整。每一件作品都精巧繁复，颇具异域风情。人类的想象力在这些扭曲的形状面前相形见绌。“自然界里也有菊石目生物这种古怪的玩意，它们看起来就好像来自外星球，它们应该是在一套不同的进化系统中进化出来的，”莱瑟姆说。“我的作品和它们一样，看起来挺‘自然’的，但其实都是组合而来。打个不恰当的比方，对我来说，这种创作方法的发明就像打开了一个潘多拉魔盒。”

乔恩·麦科马克是另一位用计算机创作的艺术家。他很清楚进化计算在创作上的威力。“想象力的桎梏一下子被打破了，”他说，“它还破除了你的偏见、成见和思维定势。利用进化式的生成过程，艺术家能与机器形成合力。这就好比一种“创造力倍增器”，无论是人类无法想象的作品，还是单靠计算机无法创作的作品，它都能创作出来。我现在都还在想，这个主意实在是太厉害、太令人振奋了。”

莱瑟姆的作品很快就在全球范围内受到热捧。他和麦科马克还专门拍摄视频，把这些稀奇古怪的形状搬上银幕，让它们动起来，再拿到各

大艺术展放映。“我都快成空中飞人了。当时东京、德国、香港、澳大利亚到处都是IBM赞助的大型展会，”莱瑟姆说，“IBM的公关机器把我的作品推广到了全球各地。”

尽管莱瑟姆的作品引起了热烈的反响，但也带来了一些意想不到的问题。“我自己都没有意识到，我拍的片子竟然会在露天狂欢派对上放映，”莱瑟姆笑道，“没想到我有那么多忠实的粉丝！”正因为莱瑟姆在露天狂欢派对爱好者中名声很响，著名乐队Shamen⁽¹⁾找上门来，希望他能为乐队做一段视频。莱瑟姆欣然同意，他在IBM的几家实验室里建立了一个巨大的“渲染工厂”（render farm），使用强大的计算机制作音乐视频。“IBM公关部的领导知道了这事儿，专门跑到伦敦找我谈话。‘能和你说话吗？’”他说，“威廉，你也知道，IBM可不能和Shamen这种乐队扯上关系。所以这次做完视频就算了，但是记住，下不为例！”

即使莱瑟姆在公众当中受到了热烈追捧，但艺术界对他的工作依然存在争议。争议的焦点在于，计算机生成的图像能否算作真正的艺术？“艺术圈肯定痛恨我的工作，”莱瑟姆坦言道，“很多人对我说，‘威廉啊，你本来可以成为一名杰出的艺术家的，唉，可惜了。’投身计算机行业让我与艺术圈生分了不少。我也理解他们的

想法。归根到底，艺术圈也是门生意，作品还是要拿来卖的，所以它讲究一个稀缺性。我用计算机一下子就弄了上亿件作品，这怎么能当艺术品卖呢？”

之所以会出现这个问题，部分原因在于，莱瑟姆剑走偏锋，他采用的方法和常规的路数不大一样。莱瑟姆并没有把计算机简单的看成一支智能的画笔，相反，他认为计算机有自己的权利，能像人一样具有创造力。“我们应该把计算机当成一名艺术家。我在‘形状生长与突变子’的程序代码中就充分体现了这一思想。”莱瑟姆说，“我所做的其实就是决策，判断计算机做出来的东西是否美观。”

在整个创作过程中，我唯一需要做的，就是把自己认为合适的基本技巧和原则告诉计算机，剩下的就留给计算机自己抉择了，莱瑟姆甚至考虑过让计算机自行判断作品是否美观。很多艺术家肯定不会喜欢这个想法。“单纯的从智力角度来说，这是最有趣的问题了，毕竟，你能用一台计算机来代替艺术家，这是多么了不起的事情”他说。但是就算计算机能真的做出自己的艺术，这是否意味着它们和人类具有相同的创造力呢？这个问题看起来简单，但回答起来并不容易。如果你用莱瑟姆的“形状生长与突变子”软件创作出了艺术作品，我们也许会说，你挺有创造

力的，用软件做出来的东西就和艺术一样。这都情有可原。可是如果把换成计算机呢？它也做出了和你不相伯仲的作品，为什么我们就不能说计算机和我们一样有创造力呢？

艺术家戈尔达纳·诺瓦科维克和莱瑟姆一样，也善于利用计算机科学创作艺术。她认为，计算机不可能拥有真正的创造力。“现阶段的技术发展条件下，计算机顶多可以模仿创造力，”她说，“创造力非常复杂，其中涉及到创作者的意图、情绪、反思还有批判性思维等等。我不觉得很快就可以有这么厉害的技术突破，说不定永远也不会有。”计算机能否创作出属于它们自己的伟大作品，同时经历与人类完全相同的创作过程？对于这种可能性，诺瓦科维克表示悲观。“为了达成这种转变，”她说，“计算机就得创造出复杂的社交系统，这样才能具备良好的哲学、政治社会学还有经济学条件基础，进而萌发艺术思想，就艺术的本质达成共识。也许有一天，计算机能做到这些，但在可预见的未来，我看不到任何的迹象。”

但是莱瑟姆与很多艺术家的想法不太一样。他坚定地认为，计算机也可以有创造力。“我个人觉得人类的创造力被高估了，”莱瑟姆说，他对“哲学意向性^{[\(2\)](#)}”、“情感内容^{[\(3\)](#)}”等艺术界常用词汇非常不以为然。“那纯粹是自我吹

嘘，”莱瑟姆笑着说，“其实都是些垃圾。搞艺术的就喜欢一天到晚神叨叨的。我从来不会说那种话。艺术不是什么神秘的东西。真正的艺术家只不过是眼光稍微比普通人高一点罢了。”

莱瑟姆也许有点太过谦虚了，毕竟，他对软件的贡献不可小视。麦科马克对此心知肚明，“说到进化艺术，这一领域的许多方法都受到了生物进化的启发，”他说，“威廉·莱瑟姆和卡尔·西姆斯都是进化艺术的前驱。理查德·道金斯（Richard Dawkins）虽然不是艺术家，但他也为这个领域贡献了不少优秀的想法。”

对于创造力这个话题，同为计算机艺术家的麦科马克持和莱瑟姆持有相同的观点：“我觉得，计算机可以独立于人类进行创作活动，”他表示，“计算机已经可以发明新的东西，计算进化系统清晰地证明了这一点。有很多创造力强的活动其实已经由计算机代劳了。比如，现代数码相机上带有智能自动模式，可以自动识别镜头里的场景，然后调整所有的参数设置，甚至连最后拍照片、按快门都不用我们操心！放到以前，摄影师只能运用自己的经验技术和创造力，才能完成摄影过程。最新版的Adobe Photoshop有一个模式，叫做智能内容填充，它的功能是从复杂的背景里自动移除或更换物体。比方说，假如你拍摄风景图片时，把地上的垃圾也拍了进来，或者

说，画面中有一条道路破坏了你的摄影构图，那么，你就可以使用智能内容填充模式将碍眼的东西去掉。这样的技术已经取代了人类的审美判断，它们使用的正是机器的“创造力”，而且很明显，这还只是个开始。我再提个更令人头痛的问题：如果说机器可以运用审美标准作出自己的判断，那么这个标准由谁说了算？一旦我们把创造力的评判工作都交给机器，这样做会产生怎样的长期影响？”

莱瑟姆在IBM工作了六年，但是当经济衰退来袭时，公司削减了大量的研究项目和经费，他也难逃被辞退的厄运。不过莱瑟姆依然选择了留在这一行，他先后创立了两家公司——1993年成立了Computer Artwork，十年后又创立了Games Audit。在努力拓展第二家公司业务的同时，莱瑟姆依然不断受邀参加各种会议，讲述他在IBM做过的工作。有一次，他做完演讲后，数学教授弗雷德里克·莱马里（Frederic Leymarie）邀请他去伦敦大学金史密斯学院（Goldsmiths College）执教。莱瑟姆即将翻开他人生中新的一页。莱马里对莱瑟姆的研究前景非常振奋，莱瑟姆本人也意识到，如果他还想从事研究工作，就很有必要找一个安定的环境。两个人一拍即合，在莱马里的帮助下，莱瑟姆接受了金史密斯学院的教职，于2007年当上了教授。

进军学术界使莱瑟姆再一次投入到研究工作中。“我们重启了‘形状生长与变异子’项目，它已经被搁置15年了，”他表示。莱瑟姆在金史密斯学院不仅见到了老同事史蒂芬·托德及其儿子彼得·托德，还结交了莱马里等一大批新同事。他们与伦敦帝国理工学院的生物信息学研究人员劳伦斯·凯利（Lawrence Kelley）和本·杰佛里斯（Ben Jefferys）合作开展了“形状生长与变异子”项目。这个项目运用莱瑟姆的“形状生长”语言实现蛋白质结构的可视化，并综合蛋白质数据库

（Worldwide Protein Data Bank，简称PDB）中的生物信息学数据，直观地展现蛋白质过去的进化过程，并预测其未来可能的进化方式。通过现有的技术手段，人们已经能够对运算出来的最终结果进行加工，并在此基础上设计全新的蛋白质结构。莱瑟姆现在可以从艺术的角度来审视科学了。“艺术家是否应该重新设计自然呢？这就是我想要探索的问题，”他说，“这让我想起了园林设计师兰斯洛特·布朗的故事，他设计的公园很漂亮，每棵树都有其固定的位置。它们既是自然景观，同时也掺杂了许多人为因素。问题是，这种经过人为设计的景观是否比原生态的自然景观更美？”

莱瑟姆或许可以说是一位特别的艺术家，不过他真正了不起的地方在于，他开创了一个全新

的流派。作为科研人员，他却以艺术家的身份探索世界，运用富于创造力的全新方法，攻克了一道又一道难题。莱瑟姆采用的并不是一般的创作媒介。他的画笔除了刷子以外，还有计算机代码和专家意见。他的画布构成了一个蓬勃发展的创新行业。连莱瑟姆自己都没有想到，他会成为游戏和艺术界的计算机应用专家。“我学到了一个道理，”他说，“有时候，你不一定要自己具备专业知识，要学会借助别人的力量。相反，如果你自信满满，以为自己已经掌握了某方面的知识，那就得当心，说不定你只是一知半解而已。所以你得找到可以合作的人，而且要让自己表现得值得信赖，这才是最关键的。”

正因为威廉·莱瑟姆等众多先驱的开创性工作，如今人们对计算机的认识和应用才发生了彻底改观。计算机已成为日常生活中随处可见的实用工具，有了它，无论是编写计算机游戏、创作音乐、制作影视剧，还是随意捣鼓设计类软件，各种各样的创作活动都可以变得轻松很多。“20世纪80年代末，我还在皇家艺术学院读书，”莱瑟姆说，“在那个年代，创作是艺术家的特权。但事实上，每个人都是很有创造力的。现在人们用Facebook都能玩出新花样。数码摄影也早就不是什么高端玩意了。看看现在有多少人在用Unity⁽⁴⁾和Garageband⁽⁵⁾吧。我觉得，是计算

机让创作活动普及了，创作不再只是少数人的特权。”

计算生物学

在伦敦大学金史密斯学院，莱瑟姆的课题是蛋白质结构的可视化，他将与生物信息学研究人员一起开展这项工作。预测蛋白质结构在当时极具挑战性，在计算机的帮助下，研究人员成功解决了这一难题。这也是计算机应用于生物学的典型范例。人的基因表达时，会生产不同的分子链，统称为氨基酸链。这些氨基酸链会自己组装折叠，形成由复杂的球体，有些区域暴露在外，有些则被隐藏在球体内部。每个球体就是一个新的蛋白质分子，不计其数的蛋白质分子构成了细胞内部乃至细胞之间相互交流的语言。蛋白质反过来会控制基因的表达，甚至决定细胞的命运。蛋白质分子的形状决定了它的功能，所以如果能根据氨基酸的序列来预测它的形状，那就再好不过了。正确的预测会帮助我们了解基因如何将遗传信息用蛋白质语言表达出来，这对开发特效药和特效酶也大有裨益。

如今，生物信息学领域的研究人员为了预测蛋白质结构，已经开发出许多功能强大的计算机技术。它们的设计思路各有不同，有的计算大分子中哪些原子更容易把它们交联在一起；有的运

用机器学习原理，从之前已经发现的蛋白质结构中寻找相似信息；有的运用遗传算法等搜索方法，结合超级计算机，比对不同可能结构的自由能^[6]，从中选择自由能尽可能低的备选方案。

当然，生物信息学的研究对象远远不止这些。事实上，有些人可能会认为，生物信息学包含的领域不止一个。早在20世纪40年代，在计算机诞生之初，科学家就已经开始借助计算机开展这方面的研究。伦敦大学伯贝克学院（Birbeck College）的安德鲁·布斯（Andrew Booth）及其导师——伟大的结晶学家约翰·伯纳尔（John Bernal）共同研究了晶体中的原子排布。为了便于研究，布斯开创性地研制了英国最早的计算设备。1946年，他制造了一台机电式计算器。接下来的两年里，他又根据冯·诺依曼结构，研制了一台继电器计算机。后来，他将这台机器重新设计成一台真空管计算机，由他的学生诺曼·基茨（Norman Kitz）在1951年制造完成。

“将现代计算设备应用于结晶学研究，这已经不是什么新鲜事了。在此之前，要想确定一个生物学分子的三维结构，需要进行数千次庞杂的计算，常用的方法是借助不同种类的机械式计算机，”生物信息学教授戴维·琼斯表示，“现如今，如果不使用尖端的计算设备，你根本没法开展结晶学的研究工作。”

当科学家首次对蛋白质和生物体的DNA进行测序时，生物信息学的革命悄然拉开了大幕。科学家所解读的，是定义生命的分子密码，这种密码已经进化了数百万年，其中蕴含的信息量大得超乎人们的想象。人类第一个完成测序的DNA基因组属于一种能够感染细菌的噬菌体，这种病毒称作Phi X 174。它的基因组里包含了11个基因，由5386个碱基（核苷酸）组成。人类基因组里的基因数在23000左右，碱基对数目更是高达三十亿。如果一个碱基对用两比特存储的话，整个基因组的数据量将在725Mb左右。这个结果听起来还不算太坏，不过，如果你想从中找到特定的数据串，尤其是考虑到部分同源染色体配对的问题（因为人的基因总是与父母存在些许差异，所以这一点必须考虑在内），那就肯定不会觉得简单了。

没过多久，研究人员对很多其他生物的物质进行了测序。他们逐渐意识到，要想判断生物体的亲疏远近，可以检测生物蛋白的相似性。但是，要想处理这种数量级的数据，研究人员需要更新更强大的计算机技术。“测序算法取得突破是在1970年，尼德尔曼（Needleman）和文施（Wunsch）两人开发出了世界上第一个可以有效比较两串序列（譬如字母串）的算法，”琼斯说，“此后，更多巧妙的算法相继问世。”

两种生物的蛋白质序列越相似，它们的亲缘关系就越近，这说明进化树⁽⁷⁾（又称系统发生树）可以被高度精确地计算出来。乔治城大学医学中心（Georgetown University Medical Center）的物理化学家玛格丽特·戴霍夫（Margaret Dayhoff）是进化树研究领域的权威。1960年，她开发了一项前沿计算机技术，可用于搜索相似的蛋白质序列，即寻找同源序列。这项技术使她成为了世界上最早从蛋白质序列数据中重建进化树的科学家。1978年，她开发了世界上第一个蛋白质进化概率模型，这个模型直到现在还在使用。

也许戴霍夫所做的最重要的贡献之一，就是建立了一个公开数据库，存储所有已知的蛋白质序列。这一资源如今成为了大型序列信息数据库的雏形。对此，琼斯表示：“我认为，戴霍夫建立的第一个生物数据库，还有一些检索数据库的早期工具，将作为重要的研究成果，为后世所铭记。”

到了1990年，随着科技的发展，科学家已经可以对生物体的全部DNA进行测序。“最初，我们只是对细菌这类简单生物的DNA进行测序，”琼斯说，“2003年，人类基因组序列图绘制成功。如今，我们已经完成了超过一千种生物的全基因组测序。在未来几年之内，我们还会对数以千计的人类个体进行全基因组测序。随着现代

生物学需要处理的数据呈指数式增长，计算技术的应用也变得空前重要，这一点丝毫不令人惊讶。”

如今，生物信息学研究人员已经掌握着数量庞大的计算工具，它们可以用来从DNA序列中识别基因，分析基因和蛋白质表达和调控，寻找肿瘤细胞基因里的突变。研究人员甚至利用这些工具，对每年发表的成千上万个生物实验报告进行了总结。生物信息学的数据库在各个领域得到了广泛应用，比如药物开发、农业害虫抗药性研究、微生物在法医分析中的应用、气候变化研究等等。琼斯对生物信息学的影响有着切身体会，因为在这个领域，他还是位成功的企业家，他与别人合作创立了Inpharmatica公司，专门从事计算工具和数据库在生物医药方面的应用工作，具体的领域包括DNA分析，蛋白质以及药物发现等等。这家公司非常成功，不断发展壮大，拥有超过一百名员工，2006年公司被药物发现公司Galapagos以1,250万欧元的价格收购。

生物信息学还在不断地发展，并为相关研究领域提供重要的数据。“现代生物信息学已经远远超出对序列数据做文章的水平，”琼斯说，“系统生物学的终极目标，是要对一整个生物体建立数学模型，这个生物体一开始可以是一个单细胞细菌，下一步就是更加复杂的大型生物

了。要实现这个目标，我们还面临着很多的挑战，这些挑战既表现在实验生物学方面，也表现在计算生物学。”

对生物系统建模并不是什么新鲜的想法。或许让人意外的是，第一个提出“生物信息学”术语的研究人员从一开始就开展了建模工作。“20世纪70年代初，我和本·赫斯泊（Ben Hesper）把生物信息学定义为对，生物系统的信息处理过程进行研究的学科，”保琳·霍吉维格说。霍吉维格是荷兰乌德勒支大学（Utrecht University）的理论生物学和生物信息学教授。“在我们看来，信息处理是生命的标志性特征之一。因此，我们认为，除了生物物理学和生物化学以外，将生物信息学独立出来，并作为一个领域进行专门研究是大有裨益的。我们也认识到，要想研究生物系统里的信息处理过程，无论是采用动态建模还是模式分析，计算方法都是不可或缺的重要手段。”

在霍吉维格看来，当研究人员把注意力集中在唾手可得的海量基因组数据时，生物信息学的视野其实变窄了。但是随着数据管理、分析以及预测工具的日渐成熟，越来越多的研究人员正在不断拓宽自己研究要点，充分利用所有的信息，以期更好地理解生物系统。“我很高兴看到这样的迹象，这说明我们研究的目的终归是要解决这些根本问题的，”霍吉维格说，“我们认为，要想

把生物系统当成信息处理系统来理解，就有必要向生物学界重新介绍、吸收那些受生物学启发的计算方法。我们现在见到的大多数生物学模型都是不充分的，它们忽略了一个重要事实，那就是，局部互作⁽⁸⁾通常会很快波及多个层面，而这些影响常常千差万别。”

“很明显，生物信息学与我们通常意义上说的‘人工生命’存在交集，”霍吉维格接着说，“人工生命这个概念从20世纪80年代末开始流行，为新学科奠基的会议其实我也参加了。与会者在研究目的和范畴的问题上分歧很大，如果只看这两条，生物信息学与生物学还是比较相似的。”

无论这个新形式的交叉研究领域是叫生物信息学，还是计算生物学、系统生物学，又或者干脆叫做人工生命，它都是要解决一些根本性的科学问题。“这门学科要解决的根本问题是‘什么是生命？’谁不会对这个问题感兴趣呢？”霍吉维格说，“研究人员着眼的是：生命是如何从非生命物质中产生的？它们是如何进化的？简单的相互作用是怎么导致如此复杂的生物行为的？”

除了霍吉维格，还有很多像她一样的研究人员正致力于攻克生物学中的很多重要问题。其中一个问题是，复杂基因组和复杂生物体是如何进化的。“进化的主要过程从达尔文时代就已经弄清楚了，那就是突变和自然选择，”霍吉维格

说，“但是，进化是一个复杂的多层次过程，DNA层次的突变首先要反映到生物的整体层面上——譬如反映到生物的形状、生理以及行为学上，只有经历了这样的层次转化以后，才会进入自然选择的过程。而且这个层次转化本身也在不断地进化。要想弄清楚进化塑造这个路线图的详细过程，以及这个路线图对进化的反馈过程，挑战性非常大。计算方法可以帮助我们获得一些重要线索，让我们一窥进化的奥妙。”

此外，互作网络中的某些基因如何调控其他基因的表达，这也是研究人员关心的问题。“我们已经在仿真模型中研究了基因调控网络的进化，”霍吉维格说，“这些模型给出了很多让人意想不到的结果。譬如，进化早期可能存在大型的基因组，基因流失则是适应性变化中的重要过程。”形态发生是生物信息学的另一大课题。所谓形态发生，就是指生物体从单细胞发育为复杂的多细胞成年形态的过程。“形态发生其实是基因调控与细胞 / 组织的物理特性精密互作的结果，”霍吉维格说，“通过使用一个简单、但却贴近实际的生物细胞模型，我们发现，事实上，精密复杂的形态发生过程可能只是相对简单的调控产生的结果。”

生物系统不仅复杂，而且时刻都在变化，从正确的数据中摸索出准确建模的方法，这活可不

简单。尽管现在我们已经可以对生物体的一小部分进行建模了，我们还是不能对整个生物体建立详实准确的模型，即便是对一个小小的单细胞细菌也做不到。但是，这一情况可能很快就会有改观。“建模的诀窍在于你必须首先收集大量的原始数据。不同条件下细胞行为的每个方面，譬如基因活化的水平，还有蛋白质的表达水平，这些都要考虑到。然后你要对数据进行整合，从简单的基因序列数据到3D影像资料都是如此，它们的来源都很分散，整合工作必不可少。最后你才能将它们纳入一个计算机模型里，”琼斯说，“很显然，如果我们能对细胞内或者是整个生物体的每个过程都进行建模，那么这项工作的影响将是非常巨大的，它不仅能深化我们对生命本身的认识，更能加强我们对抗疾病的能力。可能下一代的计算生物学研究人员会最终解决这个问题，也可能是下下一代，但是未来十年内，我们应该就可以看到惊人的进展。”

计算机医学

生物信息学涵盖的范围很广，它也从很多重要的方面深刻影响了生物学和医学的发展。但是，还有一类计算机技术几乎在每家医院都会用到，它每天都在抗击疾病的最前线与广大的医护人员并肩战斗。这就是医学影像学背后的技术。

最原始的医学影像学技术非X射线莫属。

1985年，德国物理学家威廉·康拉德·伦琴

（Wilhelm Conrad Röntgen）发现了X射线。仅仅一年以后，英国上校约翰·霍尔·爱德华兹（John Hall-Edwards）就将其用于手术诊疗。直到20世纪40年代末，X射线造影一直都是医生们窥探人体内部的主要方法。第二次世界大战时期，又有两项新技术崭露头角，引起了人们的注意。第一项就是超声波。那时候，高频声波已经被用在了坦克装甲探伤工作中，约翰·怀尔德（John Wild）是英国裔美国人，也是一名医生，他听说了这项技术后受到启发，与工程师约翰·里德（John Reid）一起开发了可以应用到医学上的超声波技术。他们用15MHz的声波照射人体，以探测人体内部的肿瘤。与X射线技术类似，直到现在，超声波依然是医学影像学里的重要技术，可以用来定期检查孕妇体内发育中的胎儿。第二项技术是闪烁造影术，医护人员可以给患者注射或是让其吞服小剂量的放射性物质，然后用特制的伽马射线摄像机跟踪放射性物质的位置，从而达到检查身体内部的目的。

这几项技术的广泛应用为现代医学造影技术的蓬勃发展打下了坚实的基础。20世纪60年代末，英国工程师高弗雷·豪斯费尔德（Godfrey Hounsfield）（此人还是全世界首台晶体管计算机

的设计者之一）冒出了一个天才的想法。当时使用的造影技术可以反映人体内不同的特征，但都有一个缺陷，那就是，它们只能做出平面的图像。如果医生想要弄清楚问题到底出现在身体的哪个具体部位，那么我们不妨从多个不同的角度拍摄连续的多个图像，这样或许可以为医生提供更多线索。可是如果能看到体内三维影像的话，诊断治疗岂不是都简单多了吗？“长久以来，人们一直在思考如何从多个角度拍摄X射线照片，”戴维·霍克斯说。霍克斯是伦敦大学学院的医学影像科学教授。“如果你把这些照片像放映电影那样进行展示的话，你就能看到人体内的三维结构了。豪斯费尔德想到了一个好办法，他想用计算机对这些照片进行三维重建。”差不多同一时间，南非科学家艾伦·科马克（Allan Cormack）对这个概念也发展出了一套理论。

这些研究的成果就是X射线计算机断层扫描，也就是我们熟知的CT技术。X射线机在功能强大的计算机指挥下，对目标进行全方位的三维扫描，得到的扫描结果可以随意旋转，人们可以从任何角度观察和分析它们。技术的进步还不仅仅体现在这一方面。现如今，核磁共振也被用于观察人体内部。人们使用大型电磁铁改变人体内原子核的状态。这些原子核会暂时的存储电磁能量，然后将其放出。但是，不同的原子存储、释

放能量的过程也略有不同，其速度存在些许差异，而这种差异是可以被检测到的。利用这个性质，人们可以对体内软组织进行高度精确的拍摄。这样我们就拥有了另一种扫描设备——核磁共振成像，即MRI。技术上的进展还有很多。闪烁造影扫描获得的图像一般是2D的，但我们可以用计算机对获得的2D图像进行处理，将其重建为3D模式，这就是正电子发射断层扫描（PET）以及单光子发射断层扫描（SPET）。

“从那时起，医学影像学的革命就开始了，”霍克斯说，“计算机先是与医学影像技术相联系，而后就是遍地开花了。”接下来的十年里，所有的扫描设备都已经电子化，超声波和X射线检测也不例外。精确度也在不断提高，有些扫描设备甚至已经达到可以检测单个细胞的水平，扫描速度的提高使得病人受到放射的时间大大缩短，副作用水平也相应降低。现在新型的成像技术还在不断涌现，为我们窥探身体内部提供了更多的选择。内窥镜就是这样一个例子，它使我们利用一个小型光学摄影机进入体内进行拍摄。“人体其实是有许多不同种类的管道构成的，”霍克斯说，“你可以将设备插入这些管道，我们通常把这样的设备叫做内窥镜。我们可以利用它来观察不同器官或结构内侧发生的事情。我们从口部进入咽喉，将仪器送入下面的管道，既

可以进入肺部的细支气管，也可以进入胃部，也就是食道。我们还可以从管道下侧插入仪器，譬如结肠或是尿道。所以我们有多种方法可以观察到身体的内部结构。这些管道的内侧之所以重要，是因为大多数的癌症都属于我们称之为上皮癌（**carcinoma**）的类型。这类癌症通常都是由组织的内侧上皮细胞突变后恶性增生而来。所以我们可以用光学技术对这些部位进行检查。时至今日，我们甚至可以吞服“摄像机胶囊”，即胶囊内镜，达到检查目的。这些胶囊会用无线传输的方式，将照片发送到体外的计算机上。而病人在服用这种胶囊后，胶囊会如正常食物一般，穿过消化道，最后被排出体外。”

如果没有计算机，取得这些进展无异于天方夜谭。计算机在幕后默默地进行了数千亿次的运算，如果没有它们的帮助，我们不可能获得如此高质量的医学扫描影像。计算机帮助我们对扫描图像进行三维重建，不仅可以把一系列图像连在一起做成电影，而且还可以根据扫描图像对整个三维结构进行计算。在此基础上，我们才能看到各个容积⁽⁹⁾的影像，放大目标器官，从各个角度观察图像细节。可以说，如果没有计算机，我们根本不可能获得这些三维影像资料。

伴随着扫描技术的进步，有个问题也越来越明显：数据太多了，多到看不完。假设你是个医

生，如果能看到病人的一系列高质量的影像和3D重建资料，这对你的诊断肯定是极其有用的，但如果要在每个病人身上都花上几个小时的时间来观看这些资料，你可能会错过一些关键性的影像，以至于无法对病人做出正确的诊断，这样就得得不偿失了。“想象一下，假如有一部电影摄像机，它每秒可以拍摄五十帧，”霍克斯说，“每帧都是一张二维图像。我们的核磁共振扫描仪也可以拍摄图像，虽然拍摄速度基本相同，但这些图像是容积图像。所以如果用像素为单位，我们可能面对的值是 $512 \times 512 \times 512$ ，每秒都要拍许多帧。产生的数据量巨大，可是放射科医生在学习的时候一次只看一张图，现在他们就要被海量数据淹没了。”

为了克服这个问题，计算机在医学影像领域发挥的作用越来越大。除了显示图像以外，计算机现在还要处理图像，并从中提取信息，在不丢失重要细节的情况下，将大量信息浓缩为较容易理解的形式，方便放射科医师和其他医生的判读诊断工作。“我们开发的一个经典工具就是图像分割，”霍克斯说，“这意味着我们可以把关键部位的图像从原始数据中抽取出来。按照字面意思，这项技术无非就是将图像分割成一个一个小部分，然后将其标记，这样就容易找到我们需要的图像。举个例子，我们用MRI可以得到一个

大脑的基本轮廓。如果我们想要对其进行标记的话，就得找到它的边界。只有将大脑的各个部位鉴别出来，我们才能对其进行测量和比较，从而判断出某个部位是否因为萎缩而变小、某个部位是不是长了个肿瘤等问题。分割技术正是达到这个目的的关键手段。”计算机视觉领域涌现出的技术也被吸收进医学影像学中，研究人员对其加以改进，大大增强了影像学的诊断能力。但是，还有一个极其困难的问题尚待解决。“相比训练有素的影像判读专业人员，计算机的诊断能力还不够，很难达到专业人员的水平。”霍克斯说，“这个领域的研究已经开展了超过二十年，现在依然十分活跃。我们正在稳步前进中。”

计算机还可以被用于不同图像间的特征匹配，这项技术就是图像配准，它对诊断学极为有用。假设一个病人在不同时间拍了两张影像照片，上面有同一个血管、器官等特征，我们对两张图像进行匹配、叠加的过程，称为图像配准。“之所以想做到这一点，是因为我们需要测量某些部位的尺寸变化，”霍克斯说，“比如，肿瘤变大了吗？血管狭窄了吗？这块脑组织萎缩了吗？你需要准确充分地证明你的推理，要能够长期跟踪这些变化。在这项任务上，现在的计算机做的比我们人类好得多。”

最初，人们也曾尝试运用计算机达到上述目

的，先是要辨别特征，譬如鼻尖或是眼眶，然后将一幅图像叠放在另一幅图像上，将两幅图像上特征的轮廓都勾画出来。问题是计算机可能根本找不到这些特征的准确位置。现在，我们有了一个更好的办法。先将两幅图像合并，然后再将单独图像的信息内容与合并后的图像信息进行比较。“我们发现，图片形态可能不同，里面包含的信息内容却是相似的，”霍克斯说，“如果同一个病人拍摄了两张头部影像照片，我们要重点比较眼睛里的某个部位。这些图像可能有不同的对比度，不同的特性，但是相同的地方在于，两张图像上都有眼睛。但两张图像没有配准的时候，你会在合并图像上看到四个眼睛。所以图像进行配准的过程，其实就是对信息的一个简化过程，将需要处理的信息量降低。从能量的角度看，我们合并图像的目的就是要将信息熵⁽¹⁰⁾降低，合并后图像的熵应该要比两张独立图像的信息熵之和要小。”

如果需要比较病人在不同的时间拍摄的两张不同图像，配准技术极其有用。它还能帮助将不同类型的扫描图像整合起来。“如果一个病人做了核磁共振、CT和超声波三种检测，要想把不同检测图像上的特征联系起来是相当困难的。因为从结构角度来看，这些图像可以互为补充，所以我们自然想把它们拿来整合一下。”霍克斯说。

可视化⁽¹¹⁾是计算机科学的一大分支，它与医学物理学的联姻也成就非凡。它可以对内脏器官进行逼真的渲染，外科医生在手术前就可以看到病人体内的实际情况。再把渲染出的图像与器官生理学相结合，也许未来的某一天，外科医生在上手术台前，就可以先在虚拟现实化的病人身上练习复杂的手术，降低实际手术的风险。可视化技术也使得腹腔镜等微创型手术的成功率大大提高，为将来推广至其他的手术类型奠定了良好的基础。“我希望能把多种医学影像技术结合起来，在实际手术前对这些数据进行加工整合，这样你就可以对病人体内的实际解剖学结构建立计算模型。你能从各个角度观察模型，甚至能让模型里的器官按照真实情况运作，譬如心脏跳动，肺部呼吸这些，”霍克斯说，“简而言之，成像技术加上计算机建模，这个组合可以为我们采用非侵入性的医学技术手段打下基础。”

计算机科学技术的分支为数众多，能在医学影像学领域找到用武之地的分支为数不少，上面提到的这些也只是冰山一角。也许未来最激动人心的事情，莫过于机器可以自己学习如何预测。我们使用常规手段筛查女性乳腺癌时，人数多，拍摄的影像资料更是数以百万计。可是只有很小的一部分人会查出癌症，约占总数的千分之五。但是数据量这么大，筛查的压力也是可想而知

的。很明显，如果能使计算机学会区分正常的乳房，和那些影像学异常的乳房，将显著提高筛查的效率。“我们可以让计算机看下正常的乳房影像学是什么样子，但是人与人之间总有差异，从表观上说，乳房的大小、外形、以及人的年龄，这些因素都会反映到影像学上。然后我们再让计算机去研究病人的影像学，”霍克斯说，“也许这只是个假设，毕竟我们到现在都还没有实现。计算机现在还不能判断女性受试者的乳房到底是不是正常，即有没有存在癌症的迹象。如果病人患有癌症，我们可以看到钙化现象，影像学上就可以看到亮点，我们还可以看到某些组织发生扭曲，这可能意味着癌症有所生长。肿瘤尺寸超过5mm时才会发生这种现象。机器学习的方法也许能帮助我们更早地发现疾病的蛛丝马迹。”

类似的方法还可以用于筛查前列腺癌，甚至是诊断和治疗痴呆症。“有些种类的痴呆症，特别是阿尔茨海默氏症，会让患者和他的亲人极其痛苦。”霍克斯说，“核磁共振改变了我们研究疾病的方式。在病理条件下，大脑特定部位的结构会发生怎样的改变？我们对这个问题已经了解了很多。譬如，海马是大脑中与记忆有关系的微小结构。在别的结构开始萎缩前，海马就已经开始萎缩了。我们可以用图像配准技术进行准确的测量。然后再用机器学习技术密切注意其他病理关

联结构的大小变化。每个人的大脑在22岁以后都会萎缩，但是有些人大脑萎缩的情况与常人不同。他们正是痴呆症的高危人群。”

胶囊内镜在肠道中磕磕绊绊地前进时，一直在不断地向外界发送照片。毫无疑问，如果胶囊摄像机没有将镜头对准可疑病灶，那么它拍摄出来的数百张照片其实并没有太大的用处。我们可以使用机器学习的方法，帮助我们从胶囊内镜拍摄的数千张照片中，自动筛选出那些医生可能感兴趣的图片，从而大大提高医生的工作效率和诊断的针对性。

计算机科学技术应用于医学之后，效果非常理想。医学扫描技术现在已经高度智能化，不仅速度快，而且图像精确，有效地推动了医学的转型和变革。我们现在可以检测许多类型的癌症，而且发现的时间比以往都要早，从而为患者争取到了更多的时间和更有效的救治。我们可以更准确地检测到动脉狭窄，从而更有效地诊断和治疗心脏疾病。卓越的医学扫描技术使所有重症内科疾病和损伤患者都受益匪浅。

创新依然是这个时代的主旋律，这一点非常激动人心。由于有大规模生产作为支撑，尖端的消费电子设备也推动了医学领域的进步。最新一代的平板计算机和手机其实就是装满传感器的强大计算机。无论是医护人员，还是患者自己，都

可以用它们来监测身体的重要指标。医生可以通过这种监测获得病人的长期生理病理数据。比起过去在出诊时临时测量的数据，长期的生理病理数据对医生的重要性要高出许多。“如果你打算去做一个医学体检，”霍克斯说，“那就只能去医院。你可能正巧碰上医生在做手术，他们也不管你心理是什么状态，就直接给你量血压，然后记下两个数字。你可能很紧张，也可能不紧张。也许一天才刚刚开始，又或者你刚刚吃了一顿大餐。反正他们只会告诉你，血压高了或是低了什么的。事情不应该是这样的！”

在计算机医学领域，需要研究的问题还有很多。“可能我们不知道的事情比知道的事情还要多一些，”霍克斯说，“很多凶险病症涉及的人群很广，而我们的影像技术却因为不够先进，在疾病的诊疗上没有什么太大的作为。”霍克斯积极推动着医学影像学技术的进步。他是科学家下海的又一个典型范例。霍克斯与伦敦其他研究机构的领军人物合作创立了IXICO公司，公司为基础影像技术研究、临床研究和影像技术的临床试验提供专业的技术支持，业务开展得非常成功。以霍克斯为代表的研究人员做了大量卓有成效的工作，推动了先进影像设备、智能计算技术和移动技术的广泛应用。也正是由于他们的努力，现代医学技术才得以不断进步，为拯救生命做出了巨

大的贡献。

计算机侦探

医学影像学是计算机科学大展神威的应用领域之一，它是一个极佳的例证，充分说明了计算机深刻影响人们的生活，为人们雪中送炭。但是，计算机不仅仅是医学奇迹的缔造者、自然科学的启蒙者、多元文化的传递者，而且还蕴藏着无限的商机。日常生活中的技术奇迹也许在你看来是理所当然的，但在金融界，它们的价值换算出来的财富可能会让你瞠目结舌。如果你是为企业家，那么先下手为强才是王道。

杰森·金登是一名白手起家的百万富豪，他表示：“20世纪80年代末是计算机科学的黄金时代，接触先进的计算机技术不再只是少数人的特权。你可以自己做实验，不用征求别人的许可，不用在实验室外排队等待，也不用忍受旁人的指手画脚。你有权接触先进的技术，可以自己编程，上机实验。”

当时正值人工智能领域的第二个冬天，金登代表着新一派的人工智能研究人员。这些人之所以从事研究，并不是为了探求智能的本质，而是因为，他们想要开发出精巧的软件来解决实际生活中的复杂问题。“我们想知道，怎样才能找到有效的方法来发挥计算机强大的运算能力，同时

巧妙地利用生物信息学算法来开创全新的局面。”20世纪80年代末，非线性随机优化算法和搜索算法突然变得时兴起来。遗传算法和人工神经网络等方法和模型相继问世，成为了研究界新的热门话题。金登表示：“新算法的不同之处在于，它们以数据处理为基础，以探索数据空间 [\(12\)](#) 为中心问题，以目标为导向，得出的结果也和以往大不一样。”金登等几名博士生正走在应用人工智能这一新兴领域的前沿。菲利普·特里莱文教授是伦敦大学学院智能系统研究小组

（Intelligent Systems Group）的领导者，他一向鼓励学生将研究工作与实业界联系起来。“我是个非常注意实用化的人，”他说，“一直都是这样。以应为目的做研究比其他方式更好。我们在学校做研究那阵子，金融界和我们走得很近，所以我们自然而然想到了与他们合作。”

研究小组的成员是几名博士生，包括乌尔·比尔盖（Ugur Bilge），康拉德·费尔德曼（Konrad Feldman），舒兰·古纳蒂拉克（Suran Goonatilake），苏克德夫·克巴尔（Sukhdev Khebbal），杰森·金登、阿奴普·曼加特（Anoop Mangat）。这个小组一开始也没什么特别的，毕竟，当时，世界各地的计算机系课题组不计其数。但由于获得研究基金并不容易，特里莱文和学校的政策都鼓励相关课题组与实业界开展合

作。很快，特里莱文的小组就因为能够获得第一手的金融数据而变得与众不同。20世纪90年代初，该课题组与银行、保险公司等大企业建立了合作关系。研究人员使用神经网络、遗传算法和模糊逻辑来寻找异常交易和金融诈骗。他们着眼于如何将这些方法组合起来，为金融业提供决策支持工具。

1993年，英国报纸《每日电讯报》（The Daily Telegraph）上刊载了一篇文章，提到了他们的研究项目，引起了外界的广泛兴趣。“那个星期电话就没有断过，”金登回忆道，“好多人跑过来打听情况，有些人更是直截了当地说，‘你们的研究成果能给我司带来很大的竞争优势，希望你们能把它保密，专门用来给我司提供咨询服务。’”由于发现了商机，研究小组的几名成员决定开一家公司，名为Searchspace Ltd。

头两年，他们一边上学，一边做一些咨询工作。特里莱文回忆道：“一开始，他们为我工作，我给他们发工资。后来他们为自己工作，我还是要给他们发工资！后来他们就自己单干了。”1995年，研究小组的成员搬进了自己在伦敦的办公室。

当时，硅谷的第一波企业家创造了令人称奇的科技成果，个人财富也一路飙升。这极大地鼓舞了Searchspace的创始人。“正所谓心有多远，

人就能走多远，硅谷的故事离你并不遥远，”金登说，Searchspace的几名年轻创业者正准备大展宏图，“我们最开始成立Searchspace的时候，就想把它做成《银翼杀手》（Blade Runner）里的泰瑞尔公司（Tyrell Corporation）。要知道，这样的公司那时候还没有呢。谁会来做这吃螃蟹的人？”金登的声音中透露出压抑不住的兴奋，“要是你能做成那样的水平，那就太厉害了。你知道吗？这就好比你在做实验，不受他人的约束，也不受资源的限制，而且实验的结果还会影响整个世界。”虽然树立了远大的理想，又有巧妙的人工智能算法在手，Searchspace的创始人能否找到合适的金主呢？对此，就连当事人心里也没谱。不过，已经有越来越多的公司与他们接触，他们也为每位新客户都开发了智能软件。很快，他们就发现，这些软件的效果十分惊人。他们做过英国的电力市场分析，为一家旅行社和一家大型酿酒厂都开发了智能网站，还受一家保险公司的委托，对洪水发生的风险性建模。此外他们还对一家大型超市的冰激凌销售做了详尽分析。

Searchspace也逐渐打响了自己的名号。但这些应用的范围太广了。他们所能做的最简单方法，就是拿出一个程序框架，这个框架可以为不同类型的应用提供分析所需的工具。竞争对手也都是这么做的，但Searchspace的眼光不仅限于

此。“我们想要做一些更直接的工作，面向商界的那种，”金登说，“后来机会终于来了。伦敦证券交易所找到我们，让我们开发金融分析软件，用来检测内幕交易和操纵市场这样的违规行为。我们决定一试。”

伦敦证交所将六个月的金融数据交给了Searchspace，看看他们能不能从中找出蛛丝马迹。如果市场操作包含有违法行为，它们是否会留下可以检测到的“电子指纹”？初步试验的结果取得了令人惊讶的成功。“依靠这六个月的数据，我们分析出了19笔可疑交易，”金登回忆道，“后来证交所对其中17笔交易进行了调查，结果正如我们所料。如果他们当初调查了剩下那两笔交易的话，应该也能发现犯罪活动的。”

“他们的态度完全转变了，”他接着说道，“以前，证交所调查违规交易时都会采用大撒网，找到线索后再顺藤摸瓜。他们会雇佣像退休警察这样的调查人员，安插在业界，让他们眼观六路，耳听八方，仔细观察交易情况。至于能否找到线索，很大程度上取决于办案人员的能力。现在，他们突然发现，‘原来线索都在数据里呢！’”金登笑了，“数据会指证到底谁在触犯法律。”

有了和伦敦证交所合作的经验，Searchspace的年轻创业者有了一个新的想法：开发一个交易监测系统，对数据进行统计学特征分类处理。他

们在系统里安置了监测模组（出于营销目的，模组没过多久就被命名为“哨兵”）。每个“哨兵”模组负责监视特定的问题，就好比负责专门事项的考核人员一般。它们分别内置了某些商用函数和特定领域的知识，从而达到行使特定功能的目的。这项技术首先被用于检测洗钱和银行诈骗。

他们的软件对银行的每一笔交易都进行了统计建模，具体来说，就是对涉及交易的每个自然人都做出了分析。“譬如，彼得·本特利在汇丰银行有五个账号，”金登说，“软件会总结出每个账号的交易特征，并做出统计特征图。统计的数据包括正常交易金额、异常金额、平均数额、支付频率、交易趋势线、单笔交易细节、交易的类型等等，所有这些都会在统计特征图上反映出来。五个账号对应着五个不同的维度，然后再加以汇总。所以你可以想象，我们面对的是这么一大堆统计数据，而且数据量每时每刻也都在增大。我们必须随时对它们进行处理、分析、归纳和维护。当然，前面所说的那些特征也在实时变化，它们与交易者的个人行为 and 遭遇有着千丝万缕的联系。”

这是个天才的想法。如果一个“哨兵”发现了任何不同于正常情况的交易，交易就会被标记为异常。如果你刚去某地度假，取了些现金，程序也有可能把它标记上。但同样的交易情况也有可

能是由于你的账户信息或是银行卡被盗导致的，也就是说，这可能是一次欺诈性交易。“这等于在交易环境里安装了非常敏锐的监听器材，可以监控各种交易活动，”金登说，“而且软件还有自动校准功能，可以进行非常精细的调整。你可以设置多个“哨兵”，每个“哨兵”分别负责监控洗钱、信用卡诈骗、借记卡诈骗、抵押贷款诈骗等等，所有的市场行为都可以纳入监控范围。譬如一个人结婚，有了孩子或是失业，这些都会导致他的个人经济状况发生变化。而所有这些事件，无论是福还是祸，它们造成的影响都躲不过我们的监测分析。”

由于结合了多种算法的优势，软件收到了很好的检测效果。“如果人工智能系统需要推向市场，那么成功的秘诀还是那句老话，兵贵精而不贵多。对于商业组织来说，通才系统并没有什么价值，但是，只要系统具备快速学习并通晓一门专业知识的能力，它就能赢得市场青睐，”金登表示，“因此，你可以把某个专业领域的深奥知识都整合到系统里。这样做的可行性要高很多。我们的反洗钱“哨兵”可是洗钱专家哦。”

Searchspace公司成为了伦敦证交所交易监测软件的唯一供应商，后来纽约证交所也采用了他们的产品。一夜之间，Searchspace就成为了全球股票交易监管的领军企业，媒体曝光度逐渐提

高。没过多久，Searchspace又为苏格兰皇家银行（the Royal Bank of Scotland）推出了反洗钱监测软件，新产品也复制了交易监测软件的发展轨迹。Searchspace凭借着风靡全球的反洗钱监测技术，又一次成为了横扫市场的领军企业。“Searchspace又一次成了吃螃蟹的人，我觉得这也催生了一个全新的行业，”特里莱文说。六成的顶级银行都成为了Searchspace的客户。Searchspace的产品在美国、欧洲、南非和日本都占据了市场的领导地位。

仅仅十年的时间里，Searchspace就从当初只有五人的小企业发展壮大，成为了拥有250名员工的大企业。在成长过程中，Searchspace一直位列欧洲最优秀的20强企业之列，同时也是英国、乃至欧洲范围内，持续增长最快的公司。“公司的成长太顺利了，”金登说，“一切都是那么的水到渠成。我们之所以能取得成功，是因为发明了一项具有实用价值的全新技术。只要将它应用于各行各业的统计分析，那用处可就大了。它从数据中发现人们难以察觉的蛛丝马迹。据我所知，这是唯一一项发现过恐怖分子金融活动的技术。”事实上，Searchspace在美国发现过一些恐怖分子资金往来的蛛丝马迹，为政府部门开展调查工作提供了线索。该公司还曾协助英国银行锁定伦敦七七爆炸案[\(13\)](#)元凶的账号。“凡是参与

恐怖活动的人，他们的交易活动一定不正常，”金登说，“事实上，早在爆炸案发生之前，他们就已经被软件监控系统盯上了，可惜只是被标记成信用卡诈骗犯而已。所以直到今天，我都不明白，既然追踪资金动态的重要性在业界早已达成共识，为什么我们不更进一步，深入挖掘一下它在反恐方面的潜力呢？”

Searchspace开创了交易及诈骗防护技术的先河，这项技术惠及了世界上的每一个国家，时时刻刻都在为交易者保驾护航。“我觉得这项技术挺有意义的，”金登说，“如今，我们生活在一个紧密互联但又匿名化的世界，基础设施非常完善。你在纽约可以毫不费力地从自动取款机中取出现金，到了香港也是这样，因为计算机可以把整个过程自动化，不需要人的参与就可以迅速摆平交易。我们的经营哲学可以概括为，该插手时就插手。我们希望做的，就是在每笔交易前，都确保这笔交易不是非法活动。说白了，就是将保护客户视为己任。”

Searchspace取得了辉煌的成功，这家计算机软件企业也成为了其他行业的经营榜样。但在公司成立十周年之际，几名创始人重新评估了现状，一致认为目前公司在技术上已经足够成熟，是时候功成身退了。“2005年，我们把公司卖掉了，套现1.48亿美元，”金登说，“在我们的眼

里，Searchspace是一家通常意义上的高科技企业。在反洗钱和金融诈骗的监测领域，我们成为了绝对的领跑者。有的时候，我们甚至觉得自己有点独孤求败的意思。我们完全可以转战其他领域，让别人继续完成未竟的事业。”直到现在，Searchspace开发的技术依然在反洗钱和反诈骗软件领域占据着统治地位。

卖掉公司后，Searchspace的几名创始人也分道扬镳，他们后来都在新的工作上取得了成功。金登毫不掩饰自己对计算机潜力的狂热期待。“举个例子，假如我们今天下午想出了一个新点子，”他说，“说不定明天就可以拿出成果，放到国际市场上销售，同时还可以在线上推广。这简直太令人振奋了！”

特里莱文对这一观点表示同意：“技术和信息的广泛传播推动了世界的民主化进程。如今，非洲偏远地区的人们也可以利用维基百科自学了。计算机已经完全改变了人类社会的面貌。”

“就单纯的技术发展而言，没有什么时刻比现在更加激动人心了，”金登说道，“对于已经身处其中的人来说，我们就站在这场变革的风口浪尖。也许这是人类有史以来最重大的变革。”

本书由“行行”整理，如果你不知道读什么书或者想获得更多免费电子书请加小

编微信或QQ: **491256034** 小编也和结交一些喜欢读书的朋友 或者关注小编个人微信公众号id: **d716-716** 为了方便书友朋友找书和看书, 小编自己做了一个电子书下载网站, 网址:
www.ireadweek.com QQ群: **550338315**

注释

(1) 苏格兰的一支电子舞曲乐队, 曲风受迷幻摇滚影响。

(2) 意向性 (英语: Intentionality) 是心灵代表或呈现事物、属性或状态的能力。简单的说, 很多心理活动是关于外部世界的, 意向性就是这里的“关于”。最初, “意向性”一词来自经院哲学。

(3) 我们将相关着情感体验的一切社会性内容称为情感内容, 而将情感的类型、强度、以及某一时期的持续性心境等称为情感状态。

(4) Unity3D是由Unity Technologies开发的一个让玩家轻松创建诸如三维视频游戏、建筑可视化、实时三维动画等类型互动内容的多平台的综合型游戏开发工具, 是一个全面整合的专业游戏引擎。

(5) GarageBand是一款由苹果公司编写的数码音乐创作软件, 是Mac的应用程序套装iLife的一部分。它允许用户创作音乐。

(6) 在热力学当中, 自由能指的是在某一个热力学过程中, 系统减少的内能中可以转化为对外作功的

部分，它衡量的是：在一个特定的热力学过程中，系统可对外输出的“有用能量”。

(7) 生物分类学家和进化论者根据各类生物间的亲缘关系的远近，把各类生物安置在有分枝的树状的图表上，简明地表示生物的进化历程和亲缘关系。在进化树上每个叶子结点代表一个物种，如果每一条边都被赋予一个适当的权值，那么两个叶子结点之间的最短距离就可以表示相应的两个物种之间的差异程度。

(8) 基因互作是指非等位基因之间通过相互作用影响同一性状表现的现象。

(9) 所谓容积，就是指被射线照射的器官体积。

(10) 熵的概念最早起源于物理学，用于度量一个热力学系统的无序程度。在信息论里面，熵是对不确定性的测量。但是在信息世界，熵越高，则能传输越多的信息，熵越低，则意味着传输的信息越少。

(11) 可视化（Visualization）是利用计算机图形学和图像处理技术，将数据转换成图形或图像在屏幕上显示出来，并进行交互处理的理论、方法和技术。

(12) 数据空间（dataspace）是与主体相关的数据及其关系的集合，数据空间中的所有数据对于主体来说都是可以控制的。主体相关性和可控性是数据空间中数据项的基本属性。

(13) 伦敦七七爆炸案，是指2005年7月7日早上交通尖峰时间，英国伦敦连环发生至少7起爆炸案，数个伦敦地铁铁路站及数架巴士爆炸，共造成56人死亡（包括四名恐怖分子），伤者逾百。

译后记

本书是介绍计算机科学史的入门级读物，书中的亮点主要体现在两个方面：一是讲述了大量生动的故事，这些故事的主人公都是计算机科学领域的天才，作者通过讲述他们的故事，直观地展现了每一次技术突破的来龙去脉，可读性很强；本书的第二个亮点表现在，作者本身是计算机科学家，他既有专业的视角，又有对读者的人文关怀：一方面，书中包含大量的专业知识，而且涉及数学、逻辑学、生物学、医学等多个领域。另一方面，纵使是极为艰涩的专业知识，作者都全部采用了形象直观的比喻和通俗的语句进行讲解，实乃难能可贵。就算是对计算机领域没有任何了解的人，只要稍加品读，都可以完全领会。可以说，作为一本通俗的科普读物，作者既考虑到了知识性，又尽了最大的努力，让读者的阅读体验轻松而流畅。

由于这本书包含大量专业的知识，同时又是入门级读物，我在翻译过程中也秉承了和作者一样的精神。在涉及到“奇偶校验位”、“ $P=NP$ ”、“域名解析”、“异配连接倾向”等重要概念的问题上，我一方面借鉴了比较可靠的资料来源（比如专业论文、课件等），确保专有名词及概念的译法不出错误，另一方面在自己弄明个中原

理的情况下，尽可能以通顺的语句表达专业领域的概念，就算是做注，也从背景资料中优先选择了较为易懂的专业解释。比如，在011章的《学习计算机编程》小节，笔者在为“复杂指令集”做注时，并未直接采用定义，因为定义较为晦涩难懂，而是直接引用了一个专业人士在讲解这个概念时打的比方。在解释“奇偶校验位”时，我适当采取了补译，从最基本的比特的概念讲起，将“奇偶校验位”的概念涉及到的全部知识难点理清楚，让读者不至于在这个地方卡壳。

本书的翻译过程历经坎坷，编辑自始至终给予了我最大的支持，也费了很大的心思，我想借此机会致以最诚挚的感谢。同时也感谢东西文库引进了如此精彩的科普读物。我希望这本书的每一句话对于读者来说都是流畅可读的，也希望我的译作能给所有想要了解计算机知识的人带来帮助。

顾纹天
2013年9月于上海

如果你不知道读什么书，

就关注这个微信号。



公众号名称：幸福的味道

公众号ID：d716-716

小编：行行：微信号：491256034

为了方便书友朋友找书和看书，小编自己做了一个电子书下载网站，网址：www.ireadweek.com
QQ群：550338315 小编也和结交一些喜欢读书的朋友

“幸福的味道”已提供120个不同类型的书单

- 1、 25岁前一定要读的25本书
- 2、 20世纪最优秀的100部中文小说

- 3、 10部豆瓣高评分的温情治愈系小说
- 4、 有生之年，你一定要看的25部外国纯文学名著
- 5、 有生之年，你一定要看的20部中国现当代名著
- 6、 美国亚马逊编辑推荐的一生必读书单100本
- 7、 30个领域30本不容错过的入门书
- 8、 这20本书，是各领域的巅峰之作
- 9、 这7本书，教你如何高效读书
- 10、 80万书虫力荐的“给五星都不够”的30本书
-

关注“幸福的味道”微信公众号，即可查看对应书单

如果你不知道读什么书，就关注这个微信号。