计算机导论整理

第一讲 什么是计算机科学和计算机

（一） 什么是计算机科学？

计算机科学是一种使用计算机来解决现实世界中的问题的一门学科。是培养计算思维并来解决一些目标问题的一门学科。

计算机科学应用的地方很广泛。

（二） 什么是计算机

1. 物理定义：

计算机时一种能够进行快速计算的自动化的机器。其计算速度几乎可以解决所有的问题。与此同时，计算机是可编程的，是可配置的，这就区分了计算机和计算器。

2. 微观定义：

根据冯诺依曼体系，计算机由三部分构成——CPU（包含运算器、控制器和若干寄存器）、内存和 I/O（输入与输出）。可以非常快速地进行存储、传递、渲染、搜索等操作来处理数据信息。

3. 计算机计算什么？

数字数据

Integer 类型

Float 类型

*二进制系统

***计算机为什么可以处理除数字以外的东西？**

原因：对数据进行编码，让数字赋有其它的含义。那么进行计算也就意味着在 IO 设备和软件的帮助下处理信息。被编码的、携带信息的数据会被 IO 设备输送给人类。

4. 计算机到底是怎么进行计算的？

布尔计算：理论基础

将许多 ANDNOT（与和非）结合起来，可以完成所有计算。

数字电路：物理实现

ANDNOT gate（与非门）

晶体管的发明：William Bradford Shockley、John Bardeen 和 Walter Houser Brattain 三位科学家于 1947 年 12 月 23 日在美国贝尔实验室研发出晶体管，并获得了 1956 年的诺贝尔物理学奖。

摩尔定律：当价格不变的时候，集成电路上可容纳的元件的数目每隔 18-24 个月便会增加一倍，性能也会提升一倍。

大规模集成电路使得计算成为一种编程。

5. 计算机的分类：

根据大小：服务器、个人电脑、笔记本、智能手机、监视器……

根据用途：通用计算机（General purpose computers）、专用计算机（Specific purpose computers）

第二讲 计算机系统是怎么工作的

（一） 二进制系统

最简单的表达形式，却可以组成很长的串。

最简单的单元计算：

1-bit 半加器：4 行，对两个输入数据位相加，输出一个结果位和进位。（2-input and (1+1)-output）

1-bit 全加器：8 行，处理低位进位，并输出本位加法进位。（(2+1)-input and (1+1)-output）

CPU 的组成：n-bit 加法器。（即 n 个加法器串联起来）

目前 CPU 可以进行的操作：

N 位加法

N 位逻辑运算

在进行特殊设计之后，CPU 可以完成：

浮点数计算

负数运算

……

或者经过添加其他的电路来完成一些特别的操作：

CISC（复杂指令集计算机/技术）：一定需要下拓展复杂的计算操作，如 Intel。

RISC（精简指令集计算机/技术）：将操作简单化，如 ARM。

（二） 如何操作计算机来解决问题——编程与程序

1. **编程的定义：**许多操作组成的一种顺序。
2. **编程的过程：**人编写指令让计算机运行。
3. **程序设计语言：**

分类：

第一代（机器语言）：机器可以理解。

第二代（汇编语言）：将低级语言抽象化使得人可以理解

第三代（高级语言）：完全面向人类。例如：Fortran, C, Pascal, Basic, C++, Java, Python, GO……

编译器的功能：将人写的指令转化成计算机能够运行的指令。

4. **程序的一生：**

程序和数据通过输入设备送入存储器。

启动程序后，计算机从存储器中取出程序指令，送到控制器中识别，分析该指令需要做什么事情。

控制器根据指令的含义发出相应的指令，将储存单元中存放的操作数据取出送往运算器进行运算，再把运算结果送回到储存器指定的单元。

当运算任务完成之后，就可以根据指令将结果通过输出设备输出。

5. **一个程序运行所经历的基础设备：**

IO 设备（输入）->内存（生存空间）->CPU（工作空间）->IO 设备（输出）

（三） 什么是软件

1. **定义：**

一种程序（Program）、算法（Algorithm）、协议（Protocol）、文档（Document）、数据（Data）（运行时）的集合。

2. **软件 VS 硬件：**

所有的程序可以被称为软件。

相对来说，所有运行程序的设备都可以被成为硬件。

没有硬件，软件就无从说起；没有软件，硬件就没有实际作用。

计算机科学所作的是设计好的软件并将其在好的电脑上运行，从而方便用户使用。

3. 软件的分类：

- 系统软件：面向开发者
  - 操作系统
  - 开发者工具（平台/环境）
  - 实用程序（Utilities）
  - 第三方库
- 应用程序软件：面向用户，解决特定的问题。

- 操作系统：最重要的系统软件。
  - 如 Windows, MacOS, 基于 Linux 的操作系统，Android, IOS, HarmoneyOS 等等。
  - 功能：操控 CPU、内存和 IO 设备，让程序能够运行。

第三讲 图灵机和 P/NP 问题

（一） 图灵机

1. 为什么需要图灵机？

图灵机为所有的计算机提供了一个统一的解决问题的模型。任何图灵机可以解决的问题在其他任何计算机上都可以实现。是计算问题的理论基础。

2. 什么是图灵机？

图灵机是一种公式化的模型，可以理解为有一条纸带，上面有排列好的 0 和 1（其中 1 的个数代表**一个十进制的数+1**，每个数之间用一个 0 隔开），还有一个指针，通过一定的操作顺序指针会向纸带的不同方向移动，并做出相应的改变。它一共有四个参数：Q, Σ, s, δ。其中 Q 是有限个状态的集合（指针），Σ 是有限个符号的集合（指针所做的事），s 是最初的状态（$s \in Q$）（指针指向的数字），δ 是变换方式，并且决定下一个步骤（指针应该进行什么样的操作，然后往哪边移动）。

3. 图灵机如何运作？

下面举一个实现两个数字相加的图灵机的例子。相加的图灵机的操作表格如下：

TABLE 4: Transition table for T_{Add_2}

	0	1
q_1	/	$0\,R\,q_2$
q_2	$1\,L\,q_3$	$1\,R\,q_2$
q_3	$0\,R\,q_4$	$1\,L\,q_3$
q_4	/	$0\,R\,q_{\mathrm{halt}}$

将上面所说的符号与上图对应就是：

TABLE 4: Transition table for T_{Add_2}

	0	1
q_1	/	$0\ R\ q_2$
q_2	$1\ L\ q_3$	$1\ R\ q_2$
q_3	$0\ R\ q_4$	$1\ L\ q_3$
q_4	/	$0\ R\ q_{\text{halt}}$

大致讲一下意思：里面的 L 代表向左移动，R 代表向右移动。以状态 q2 为例。当状态为 q2 的时候，观察此时指针指向的数值，如果指向的是 0，那么就把该指针指向的这个数字变成 1，然后把指针向左移一格，然后将状态变为 q3；如果指向的是 1，那么就把该指针指向的这个数字变成 1（也就是不改变数值），然后把指针向右移一格，然后将状态变为 q2。

比如说现在要实现 3+4，那么在纸带上初始情况就是这个样子：

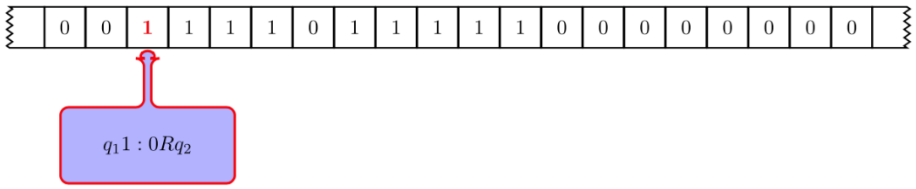


FIGURE 4: The computation of 3 + 4 by T_{Add_2}

现在指针指向第一个 1，并且状态为 q1，那么根据上表，我们应将该指针指向的数值变为 0，然后向右移动一格，再把状态调整为 q2。也就是变成这样：

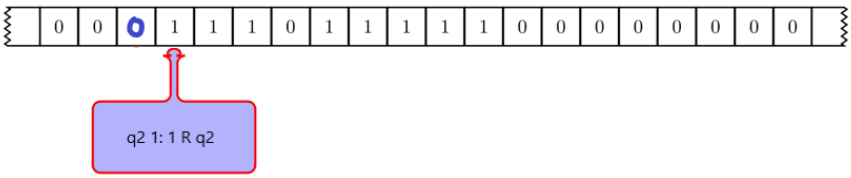


FIGURE 4: The computation of 3 + 4 by T_{Add_2}

现在指针指向 1，状态为 q2，那么根据上表，我们应该将该指针指向的数值变为 1（也就是不变），然后向右移动一格，把状态调整为 q2（也就是

状态不变）。以此类推。最后得到的结果以及指针停在的位置为：

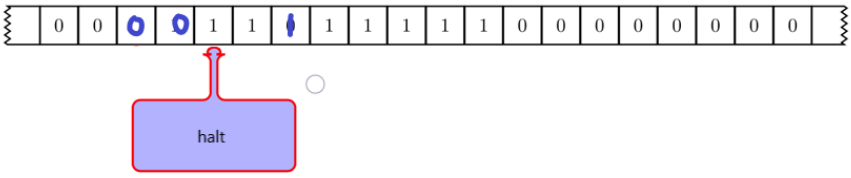


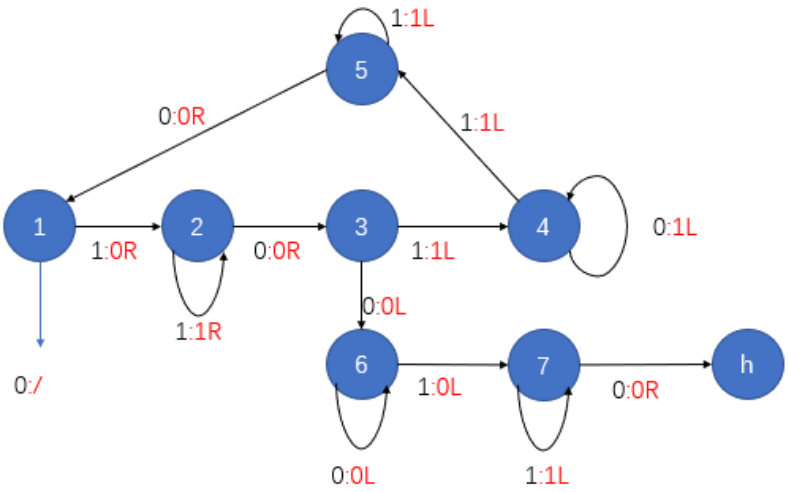
FIGURE 4: The computation of 3 + 4 by T_{Add_2}

据此拓展，这样设计图灵机可以完成 n 个数之和+1 的加法：

TABLE 5: Transition table for T_{Add_i}

	0	1
q_1	/	$0\,R\,q_2$
q_2	$0\,R\,q_3$	$1\,R\,q_2$
q_3	$0\,L\,q_6$	$1\,L\,q_4$
q_4	$1\,L\,q_4$	$1\,L\,q_5$
q_5	$0\,R\,q_1$	$1\,L\,q_5$
q_6	$0\,L\,q_6$	$0\,L\,q_7$
q_7	$0\,R\,q_{\text{halt}}$	$1\,L\,q_7$

可以用另一种示意图表示：



4. 图灵机对计算机科学的影响

图灵机对计算机科学的理论方面、现代计算机以及编程理论方面均有一定的影响。

如在计算机科学的理论方面，它是算法、计算、物理计算（Physical computation）、有效计算（Efficient computation）等的理论基础。

同时，图灵机也定义了什么问题计算机能解决，什么问题计算机不能解决。

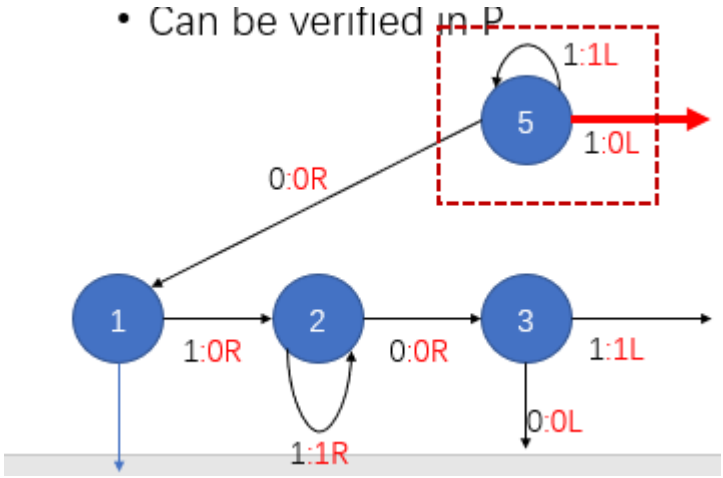
（二） P 问题与 NP 问题

1. P 问题

计算机可以解决的问题，具体说就是可以在多项式时间复杂度里、在确定的图灵机上解决的问题。

2. NP 问题

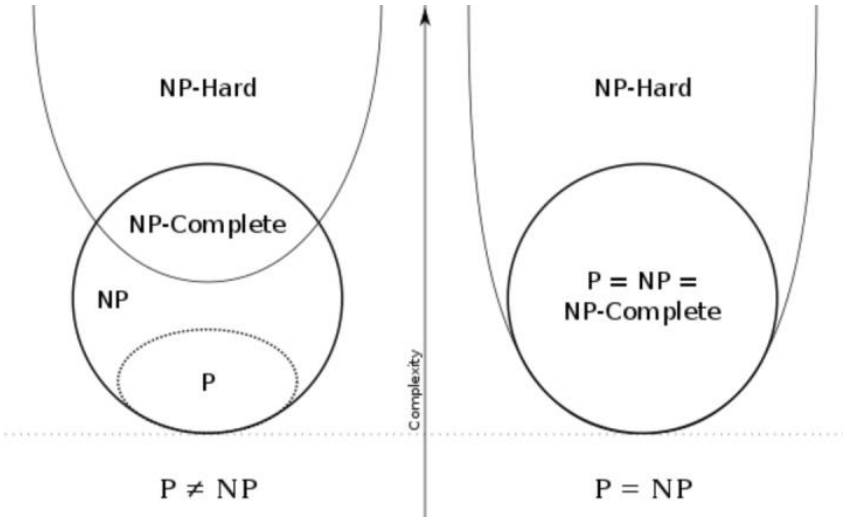
可以在多项式时间复杂度里、在**不确定的**图灵机上解决的问题。NP 问题可以被 P 问题证明。在图灵机上可以这么表示：



也就是说对于一个输入会出现多种输出。

- NP 问题的分类

- NP-hard：能被转化成 P 问题的 NP 问题以及能够在多项式时间复杂度内归约到的问题。举例：Halting problem
- NP-complete：最不可能被简化为 P 问题的问题。举例：SAT，算 24 点的完全版本



（一） 计算思维

1. 定义：

在解决问题的时候的一种高级的、认知性的科学的（不仅是但主要是计算上的）思维。

2. 解决问题时的一般过程：

现实问题—[模型化]—>问题模型<——解决所需技术

|

|----------|----------|

面向数学　面向计算机　面向…

3. 计算步骤：

- 分析
- 抽象化
- 实现并进行测试
- 评估
- 总结与推广

4. 计算思维的特征：

- 分解
- 懂得如何认知模式（Recognize patterns）
- 抽象思维
- 算法

5. 计算思维与计算机技术之间的关系：

计算思维是世界观方法论，计算机是环境，计算机技术是解决的问题和工具。

计算思维是一种概念，不是一种编程；是必要的基础，不是机械的技能；是人的一种思维方式，不是计算机的；是数学思维与工程思维的结合；是想法，不是人工制品。

总的来说计算思维是逻辑思维、算法思维、高效率思维、科学思维以及创新思维的结合。

（二） 算法

1. 定义：

一种有限的，对解决一类特殊类型的问题而给出的有序的操作的法则集合。

2. 重要的特征：

- 有穷性（Finiteness）
- 确切性（Definiteness）
- 输入项（Input）
- 输出项（Output）
- 可行性（Effectiveness）

3. 计算机科学与算法：

计算机科学的一个重要部分是算法的艺术。

编程的艺术——设计一个优雅的算法；让它们能在电脑上高效地运行。

4. 如何评价一个算法？

- 正确性
- 效率

- 空间复杂度
- 时间复杂度
  - 最佳情况（最小耗时）[Θ]：由“最简单”的输入决定，为所有的输入提供了一个最佳目标。
  - 最坏情况（最大耗时）[O]：由“最复杂”的输入决定，所有的输入不会超过这个时间。
  - 平均情况（随机输入耗时期望值）[Ω]：需要一个“随机”输入的模型，可以作为一个预测结果。

5. 为什么要分析算法？

- 预测程序的运行结果
- 比较不同的算法
- 为程序运行提供保障
- 理解理论基础
- **主要：避免程序出错

6. 储存空间大小单位及换算：

bit：一个二进制位 0 或 1

byte：1byte=8bit

MB(megabyte)：1MB=2^{20}bytes

GB(gigabyte)：1GB=2^{30}bytes

第五讲 云计算

（一） 计算机网络

1. 定义：

将多个计算机 A，B…连接起来组成一个虚拟的计算机 C，这个虚拟计算机 C 可以获取属于 A，B…计算机的资源。

2. 计算机技术与电子通信的交叉：

- 资源子网络：主机、终端、适配器、软件、数据
  数据处理
  个人化、面向用户
  基于公路网的商业化
- **通讯子网络：**通讯控制器、电路、通讯设备
  数据通讯
  共享、公共设施
  公共公路网

3. 不同角度下网络的分类：

- 地理角度：LAN（局域网）、WAN（广域网）、MAN（城域网）
- 交换角度：线路交换、信息交换、包交换、信元交换
- 通讯媒介角度：无线、有线、光纤
- 拓扑学角度：总线（bus）、星形总线（start）、令牌环（token ring）、树形（tree）……

4. 总体网络模型：

TCP/IP	五层模型	OSI/RM	功能作用	协议	数据单元
应用层	应用层	应用层	为应用程序提供服务	HTTP, FTP, TFTP, SMTP, DNS, TELNET, HTTPS, POP3, DHCP	APDU
		表示层	数据格式转化、加密	JPEG, ASCII,, DECOIC、加密格式等	PPDU
		会话层	建立、管理和维护会话	TCP, UDP	SPDU
运输层	运输层	传输层	管理端到端的连接	ICMP, IGMP, IP(IPV4 IPV6), ARP, RARP	段
网际层IP	网络层	网络层	IP地址和路由选择	/	包
网络接口层	数据链路层	数据链路层	建立逻辑连接、进行硬件低素质寻址、差错校验等功能	/	帧
	物理层	物理层	建立、维护、断开物理连接	/	bit

发送端从应用层到物理层一层层打包，接收端从物理层到应用层一层层拆包。

5. 典型的网络服务（基于 TCP/IP 的应用）

浏览器、http-url 媒介应用、电子邮件、ftp、远程登录、微博、微信、支付宝……

（二） 云计算

1. 定义：

- 云：巨大的；远程的；松散的
- 云计算：将联网的计算机打包成一个云，像本地服务器一样提供统一服务，但是成本很低。不过速度有点慢。

2. 云网络：

对云服务器的网络管理、控制以及数据连接。例子包括用基于云的网络控制器通过 WAN 连接直接访问（direct traffic）；云服务提供者内部的 WAN 带动客户的 WAN 流量（WAN traffic）。

** 网络是云计算的载体。

3. 为什么云计算服务成本很低？

- 共享联网的计算机
- 集中的维护与保养，微小的实用程序
- 时间上共享网络资源，用户没有同时使用计算资源。

4. 云计算服务：

- IaaS：基础即服务——将硬件设备等基础资源封装成服务供用户使用。
- PaaS：平台即服务——提供用户应用程序的运行环境。
- SaaS：软件即服务——将某些特定应用软件功能封装成服务。

5. 云计算的特征：

- 大规模
- 虚拟化
- 可靠性强
- 通用化
- 可拓展性
- 请求式服务
- 廉价

6. 成功的云计算的例子：

AWS(Amazon Web Services), Azure(微软), Google cloud, Aliyun(阿里云)

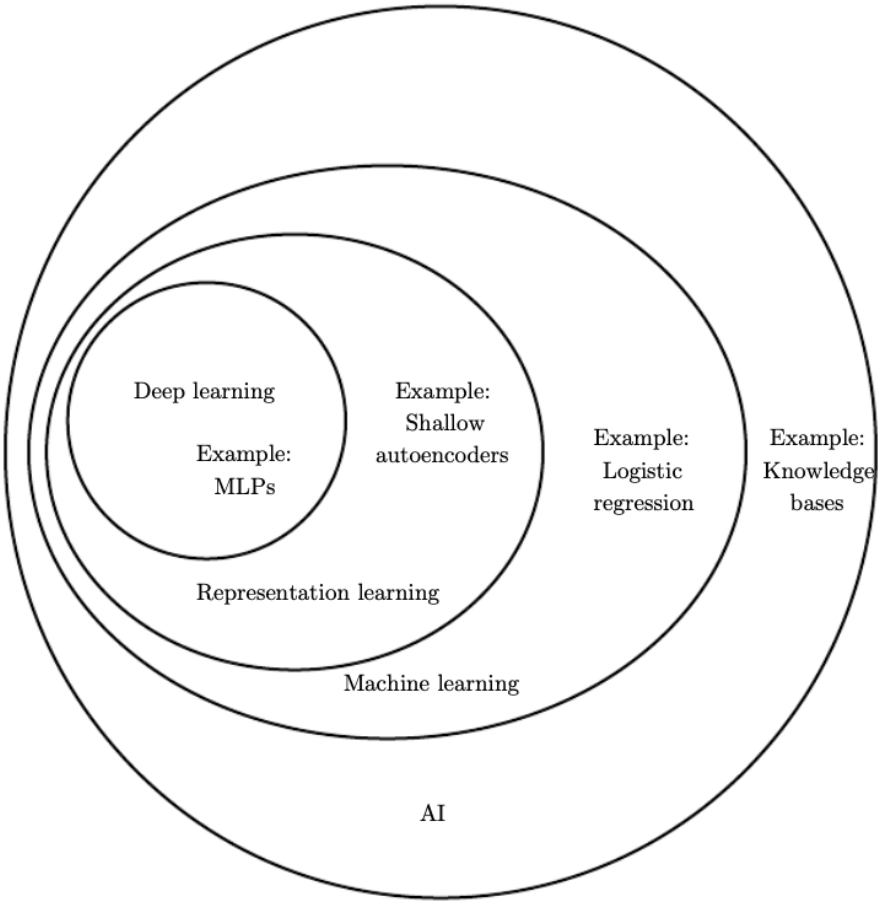
7. 云计算与大数据：

云计算为大数据提供了处理的平台。

（三） Docker

一种在云计算下递送与运行软件的方式，将不同功能的软件打包成标准化外形的映像文件。Docker 能够将应用程序与基础架构分开，从而可以快速交付软件。

第六讲 机器学习与人工智能简介



（一） 机器学习（Machine Learning）与深度学习（Deep Learning）

1. 机器学习定义：

计算机通过分析任务和性能测量获得经验，并通过这个经验提升能力。

2. 典型的机器学习的任务：

- 分类
- 聚集

3. 深度学习：

本质上是一种三层或更多层的神经网络。这些神经网络尝试模拟人类大脑的行为，使得它能从大量的数据中得到学习，尽管和人类大脑相差很远。过程中的特征是自动学习的。

（二） 人工智能（Artificial Intelligence）

1. 定义：

- 智能：人类的行为。
- 人工：是人造的，人教机器来表现出智能。

2. 人工智能四个方面的角度：

- 像人一样思考

  认知模型
- 理性的思考

  思考的规则——什么是对的，什么是错的。哲学观念。
- 像人一样的行为

  图灵测试

- 理性的行为

做正确的事情——期望利益最大化。思考为了理性的行为。

3. 理性主体（Rational agents）

一个感知和行动的实体。抽象来说，是一种从历史感知到行动的函数。

对于任意给定的环境和任务类，我们寻找有最佳结果的主体。

注意：由于计算方面的限制，完美的理性化是无法实现的。只能对给定的机器资源设计最佳方案。

4. 为什么人工智能这么难？——一门跨领域学科

结合哲学、数学、经济学、神经学、心理学、计算机工程、控制论和计算机控制学、语言学而成的一门学科。

现在的人工智能主要以计算机为基础，但不局限于计算机。我们需要包含更多的“机器”来发现（输入），决定（推断）和控制（输出）。

第七讲 大数据简介

大数据——一种新兴领域

（一） 方法：以计算为基础

（二） 处理对象：大数据

1. 什么是数据？

代表现实世界中的东西的符号。

2. 什么是大数据？——4V 模型

- 大体量（Volume）【数据的规模】
- 时效性（Velocity）【数据流的分析】
- 多样化（Variety）【数据的不同类型】
- 准确性（Veracity）【数据的不确定性】

3. 怎样算大数据？

大数据的大不是指一个固定的、绝对的大小。相对而言指的是超出一般单独一台计算机处理能力的数据规模。

4. MapReduce 编程模式——一种大数据处理框架

- 映射器（Mapper）：作用在子任务的节点上
- 归约器（Reducer）：收集映射器的结果组成目标任务的最终结果。

通过 Map 映射将任务进行分解并分配；通过 Reduce 映射将结果归约汇总输出。

5. 大数据的挑战与机遇：

- 挑战
  - 体量方面：分布式存储与 CPU
  - 种类方面：新的数据建模技术
  - 时效方面：新的算法
  - 准确性方面：新的建模技术
- 机遇：无法想象

？第八讲 AlphaFold 简介

？第九讲 如何学好计算机科学

** 计算机科学的范围：计算机工程（Computer engineering）[EE]，计算机科学（Computer science），网络安全（Cybersecurity），信息系统（Information systems），信息技术（Information technology），软件工程（Software engineering），数据科学（Data science），人工智能（AI）

2000年	姚期智（Andrew Chi-Chih Yao）	计算理论，包括伪随机数生成，密码学与通信复杂度
2001年	奥利-约翰·达尔（Ole-Johan Dahl），克利斯登·奈加特（Kristen Nygaard）	面向对象编程
2002年	罗纳德·李维斯特（Ronald L. Rivest），阿迪·萨莫尔（Adi Shamir），伦纳德·阿德曼（Leonard M. Adleman）	公钥密码学（RSA加密算法）
2003年	艾伦·凯（Alan Kay）	面向对象编程
2004年	文特·瑟夫（Vinton G. Cerf）,罗伯特·卡恩（Robert E. Kahn）	TCP/IP协议
2005年	彼得·诺尔（Peter Naur）	Algol 60语言
2006年	法兰西斯·艾伦（Frances E. Allen）	优化编译器
2007年	爱德蒙·克拉克（Edmund M. Clarke）,艾伦·爱默生（Allen Emerson）,约瑟夫·斯发基斯（Joseph Sifakis）	开发自动化方法检测计算机硬件和软件中的设计错误
2008年	芭芭拉·利斯科夫（Barbara Liskov）	编程语言和系统设计的实践与理论
2009年	查尔斯·萨克尔（Charles Thacker）	帮助设计、制造第一款现代PC

2010年	莱斯利·瓦伦特（Leslie Valiant）	对众多计算理论所做的变革性的贡献
2011年	犹大·伯尔（Judea Pearl）	人工智能
2012年	莎菲·戈德瓦塞尔（Shafi Goldwasser），希尔维奥·米卡利（Silvio Micali）	由于在密码学和复杂理论领域做出创举性工作
2013年	莱斯利·兰伯特（Leslie Lamport）	在提升计算机系统的可靠性及稳定性领域的杰出贡献
2014年	迈克尔·斯通布雷克（Michael Stonebraker）	对现代数据库系统底层的概念与实践所做出的基础性贡献
2015年	惠特菲尔德·迪菲（Whitfield Diffie），马丁·赫尔曼（Martin Hellman）	非对称加密的创始人
2016年	蒂姆·伯纳斯·李（Tim Berners-Lee）	万维网的发明者
2017年	约翰·轩尼诗（John Hennessy），大卫·帕特森（David Patterson）	开发了RISC微处理器并且让这一概念流行起来
2018年	约舒亚·本希奥（Yoshua Bengio），杰弗里·欣顿（Geoffrey Hinton），扬·莱坎（Yann LeCun）	人工智能深度学习方面
2019年	Patrick M. Hanrahan，艾德文·卡特姆（Edwin E. Catmull）	对3D计算机图形学的贡献，以及这些技术对电影制作和计算机生成图像（CGI）等应用的革命性影响
2020年	杰弗里·戴维·乌尔曼（Jeffrey David Ullman）、阿尔弗雷德·艾侯（Alfred Vaino Aho）	创造了全球数百万编程人员使用的工具和教材，推进编程语言实现的基础算法和理论，并在极具影响力的书籍中综述了这些研究成果