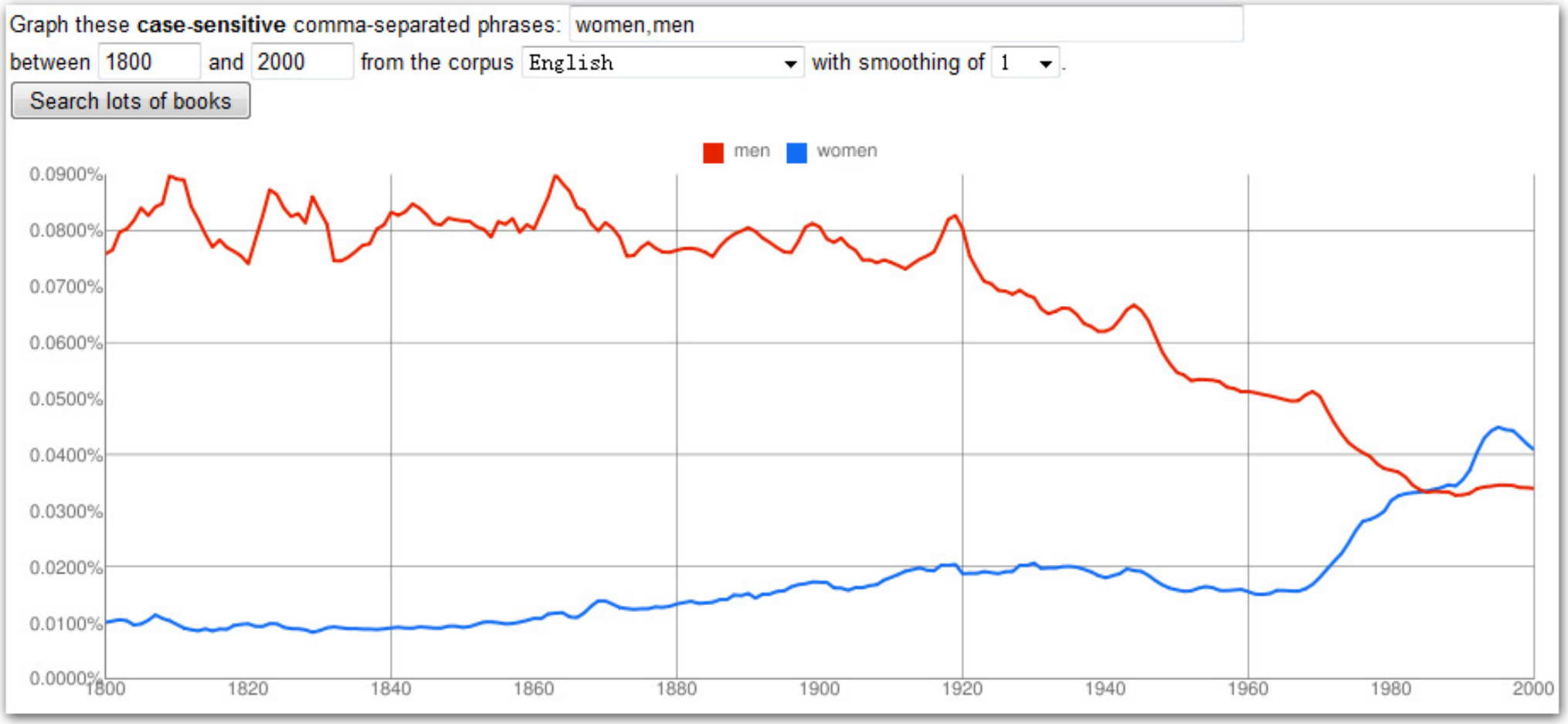
数据科学与工程导论

Introduction to Data Science and Engineering

Google对书籍的处理不仅是扫描，还进行了数字化(Digitizing)与数据化(Datafication)，庞大的书籍数据库甚至孕育了一个新学科的成立：文化组学(Culturomics)。

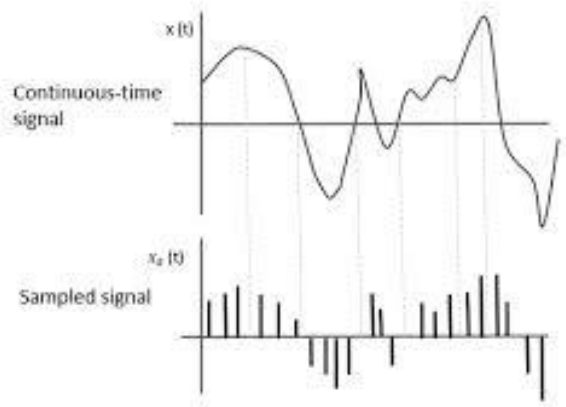
Google 的数字化

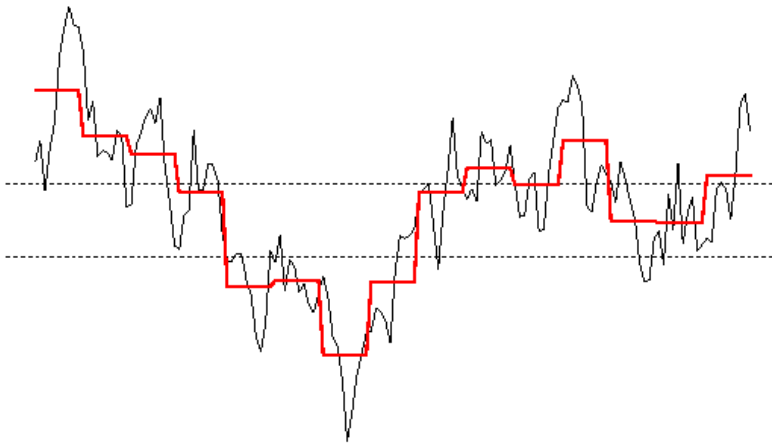
开篇实例



Google Books Ngram Viewer

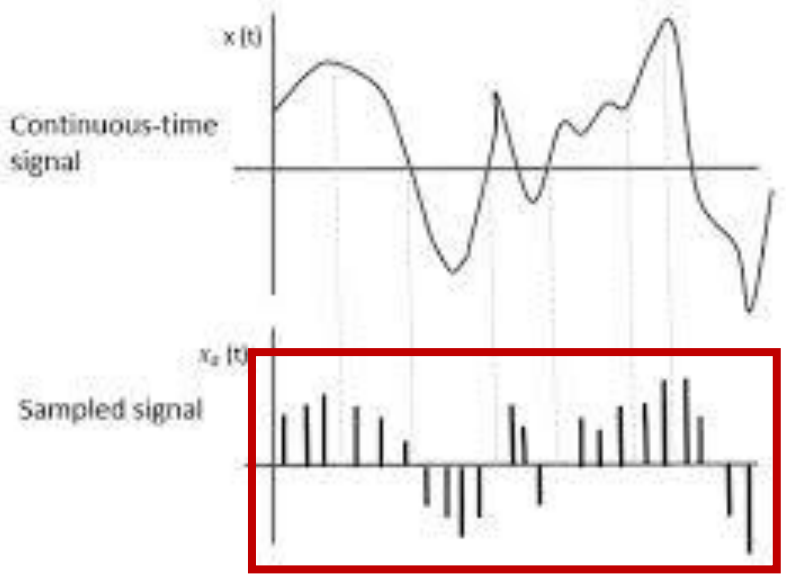
第 3 章
数据的模型与结构



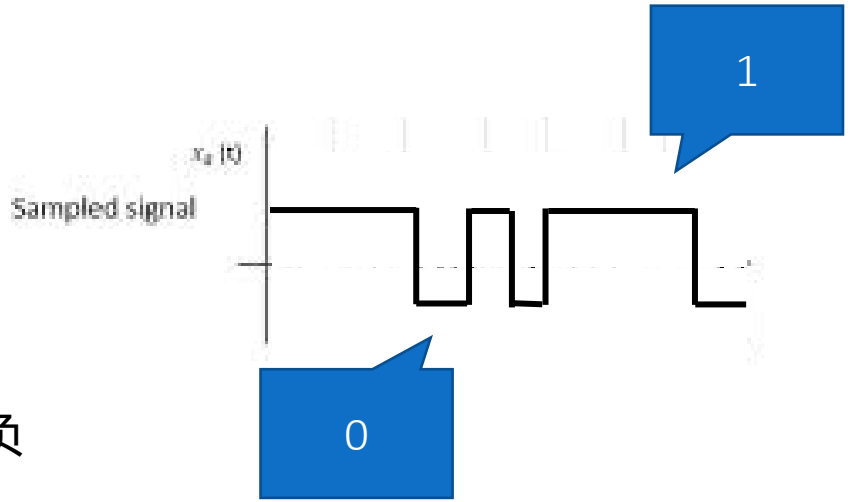


连续信号(模拟信号) → 离散信号(数字信号)

数据的离散化



仅表示正负



数据的离散化

010111011001…

101110100101…

00110010 00101101
11001100 01011011
……

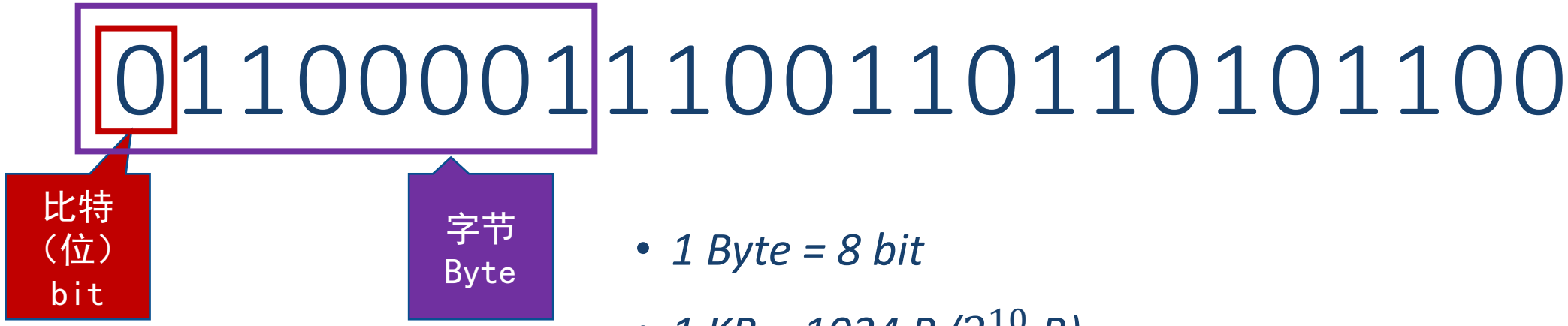
00110 0001 00001000
10100 0010 00000001

ADD AL,08H
MOV BX,AX

离散化的目的：表示数据、存储数据、处理数据

数据的离散化

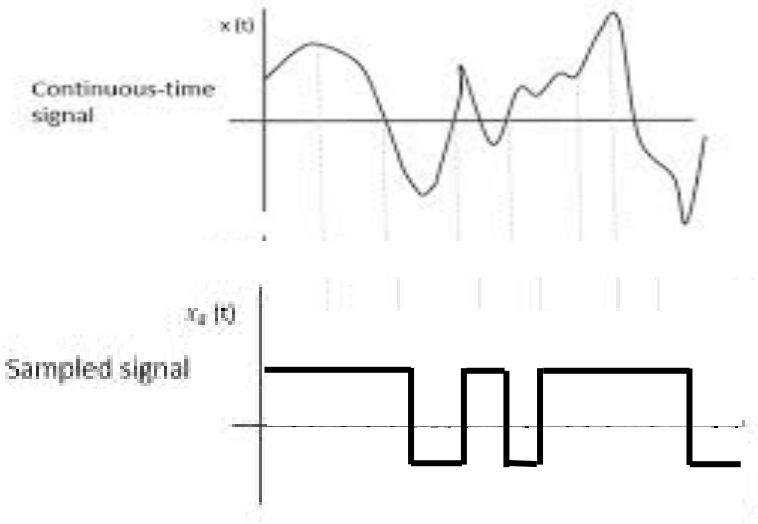
- 计算机中使用二进制表示数据、存储数据、处理数据
- 二进制：0 1

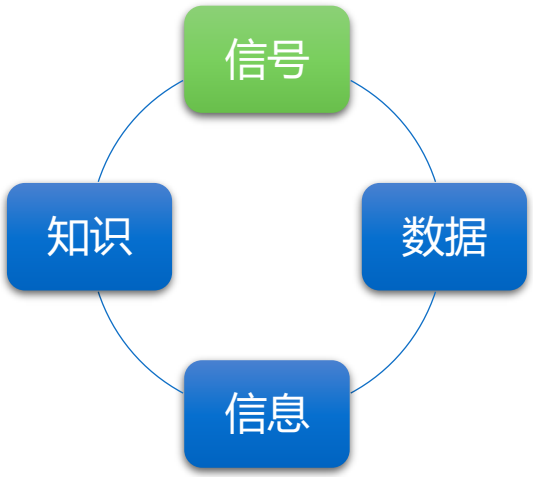


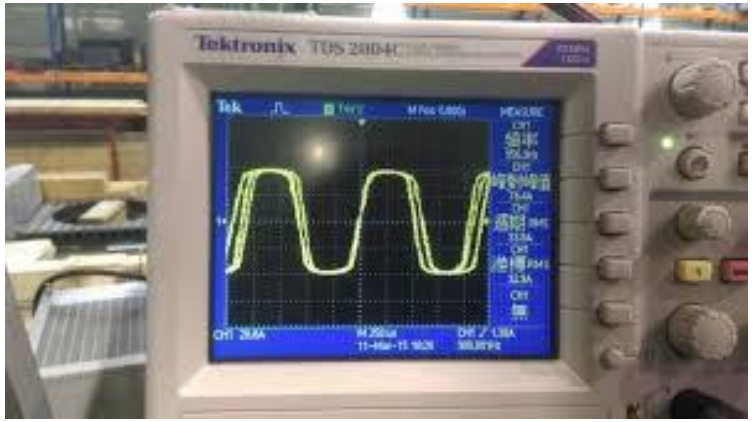
- *1 Byte = 8 bit*
- *1 KB = 1024 B (2^{10} B)*
- *1 MB = 1024 KB*

二进制与比特

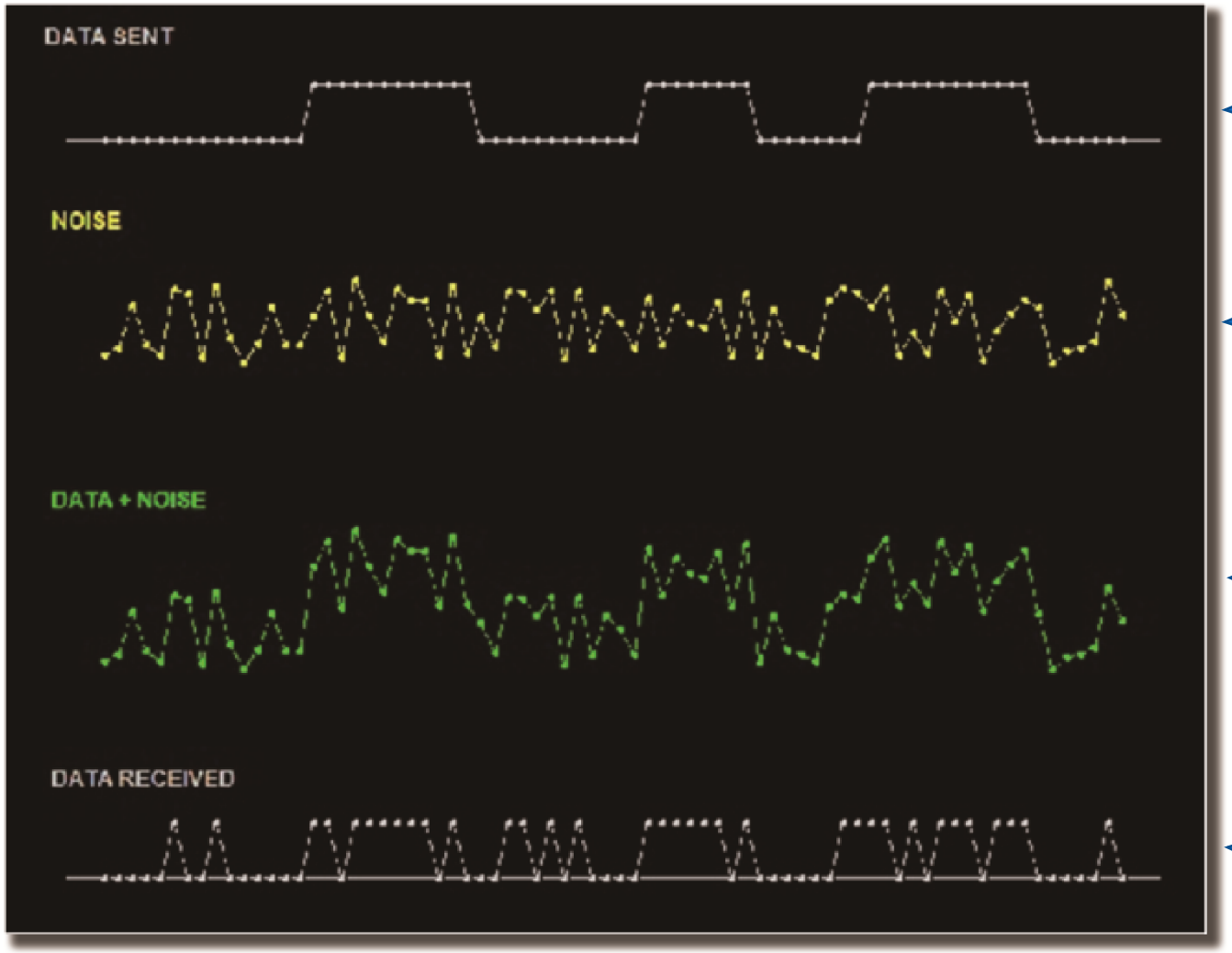
- 数据传输过程中以电磁波的表示形式
  - 包括电信号、光信号等
- 分为模拟信号和数字信号







信号、数据、信息和知识

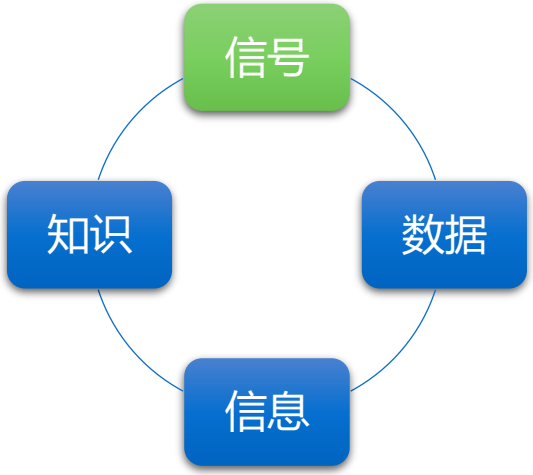


原始信号

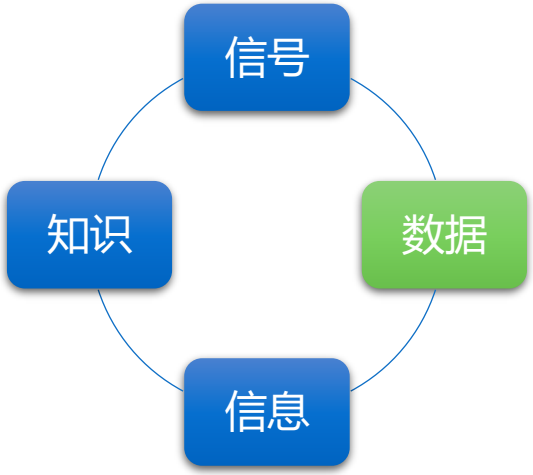
噪声

叠加后实际传输的信号

接收端解码后得到的信号



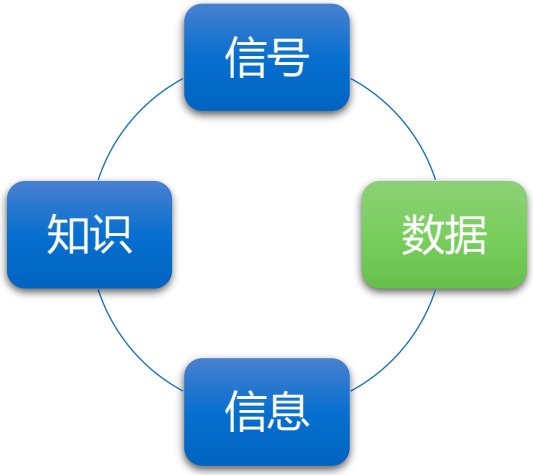
信号、数据、信息和知识



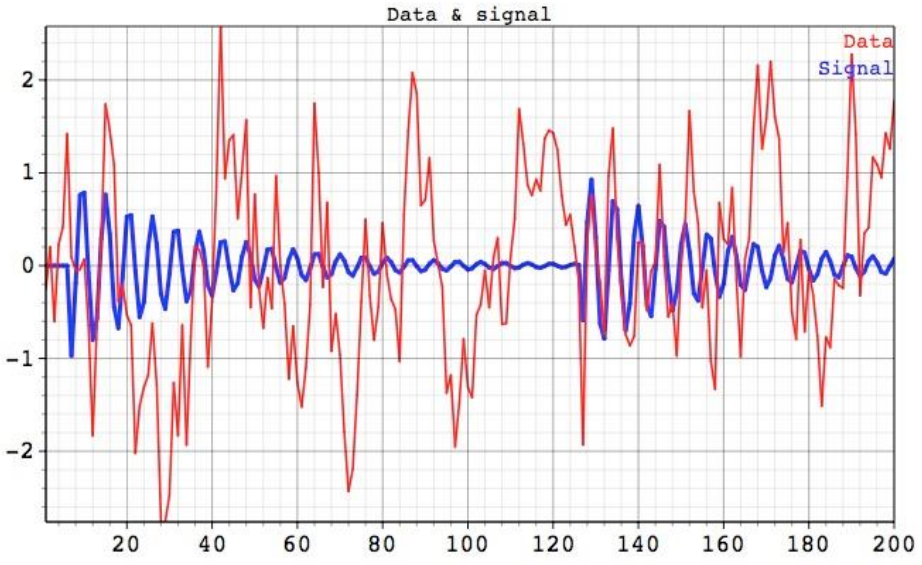
- 很多人认为，数据就是数字，或者必须是由数字构成的，其实不然，数据的范畴比数字要大得多。
  - 互联网上的任何内容（文字、图片和视频）；
  - 医院里包括医学影像在内的所有档案也是数据；
  - 公司和工厂里的各种设计图纸；
  - 出土文物上的文字、图示，甚至它们的尺寸、材料；
  - 宇宙在形成过程中的许多数据，如宇宙基本粒子数;
  - 人类活动本身。

数据的范畴是随着文明的进程不断变化和扩大的。

数据的范畴



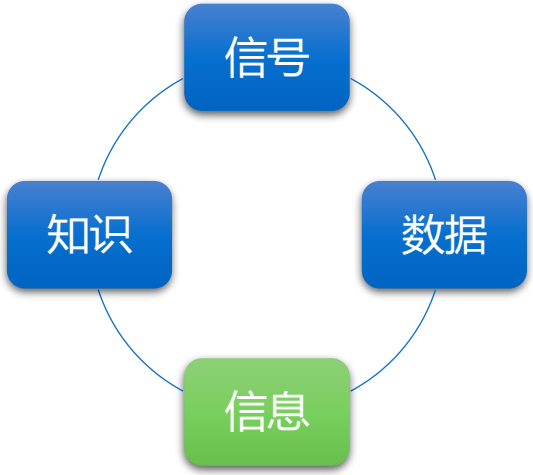
- 数据本身是人造物，因此它们可以被随意制造，甚至可以被伪造。没有信息的数据通常没有太大意义。
  - 例如，优化网页搜索排名而人为制造出来的各种作弊数据
- 数据常和毫无意义的数据和伪造的噪音混在一起；需要过滤掉没有用的数据，从而获取数据背后的信息。

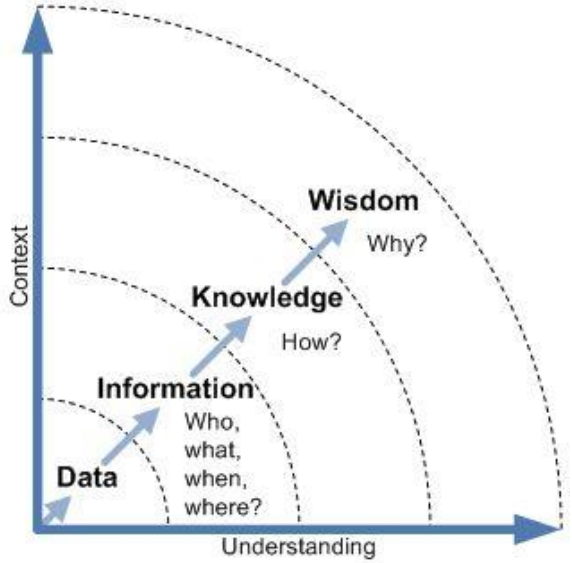


数据与噪音

- **信息**是具有时效性的有一定含义的、有逻辑的、经过加工处理的、对决策有价值的数据流。

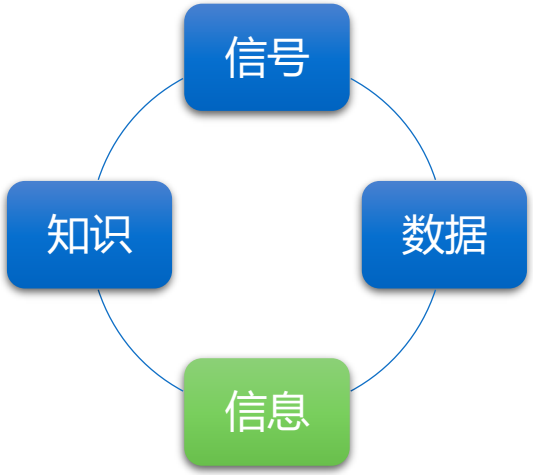
- 数据中隐藏的信息和知识是客观存在的，但是只有具有相关领域专业知识的人才能将其挖掘出来。



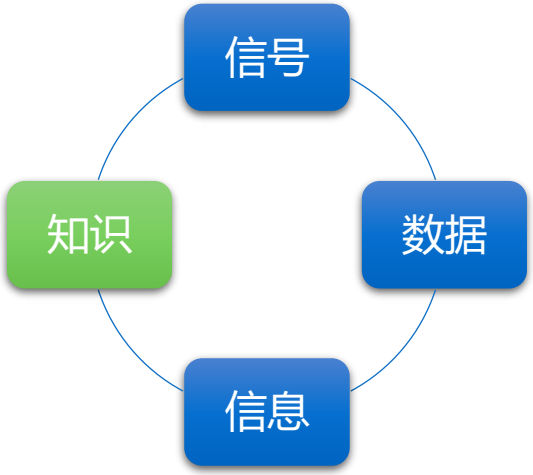


信息的含义

- 人们常常把数据和信息的概念混同起来。
  - E. g. 数据处理和信息处理
- **信息**是关于世界、人和事的描述，它比数据来得抽象；而数据则是信息的载体。
- 信息既可以是人类创造的，也可以是天然存在的客观事实
  - E. g. 两个人的语音通话记录，地球的面积和质量
- 信息有时藏在事物背后，需要测量和挖掘才能得
  - 宇宙大爆炸的证据、物理学定律中的参数、日月星辰运行的周期



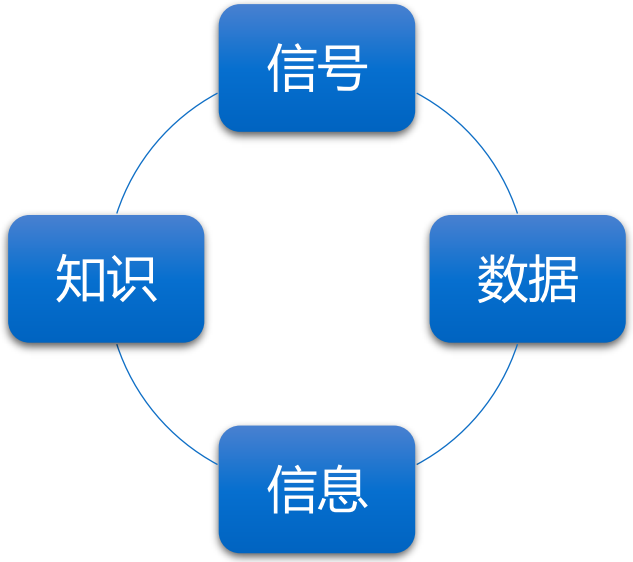
数据和信息



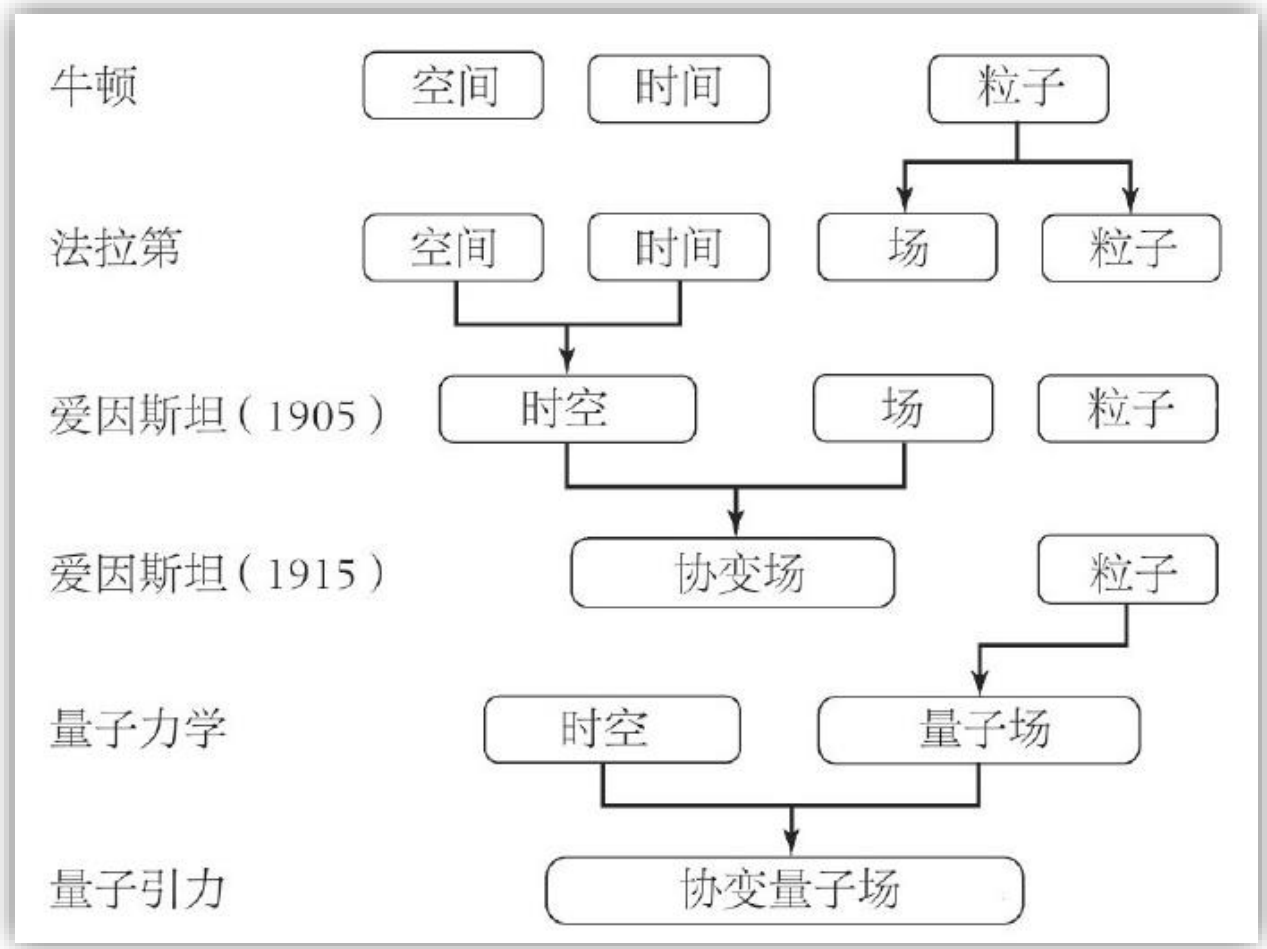
- 知识比信息更高一个层次，也更加抽象，它具有系统性的特征。
  - 比如通过测量星球的位置和对应的时间，就得到数据；通过这些数据得到星球运转的轨迹，这就是信息；通过信息总结出开普勒三定律，就是知识。
- 人类的进步就是靠使用知识不断地改变我们的生活和周围的世界，而数据是知识的基础。

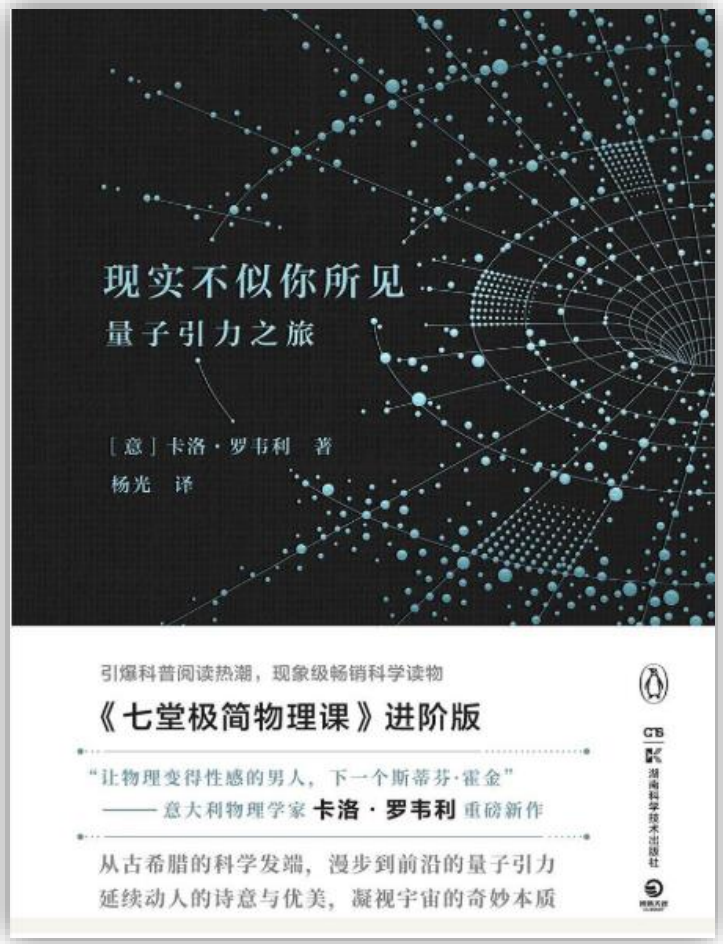
数据和信息

- 从现象、数据到信息、知识，抽象层次是越来越高的。
- 知识的抽象层次是很高的了，而知识中抽象层次最高的，应该就是基础概念。因为这些概念是知识大厦的基石。
- 抽象层次和处理数据的能力，也都是衡量文明发展程度的重要标准。



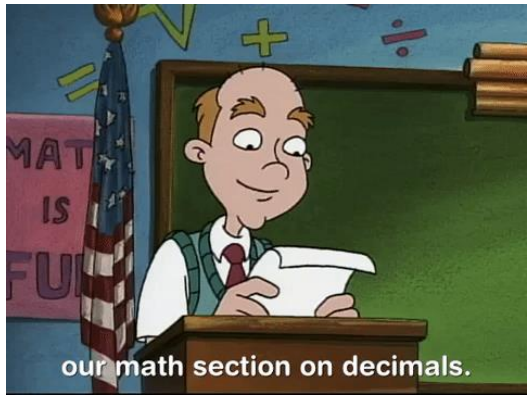
从现象到知识

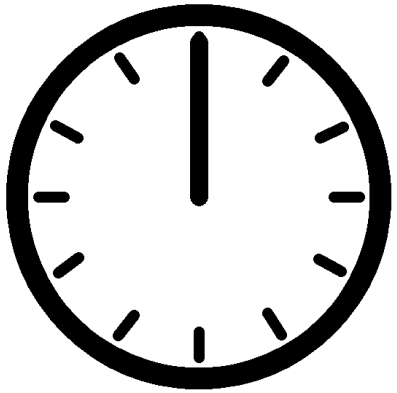

牛顿
空间
时间
粒子
法拉第
空间
时间
场
粒子
爱因斯坦（1905）
时空
场
粒子
爱因斯坦（1915）
协变场
粒子
量子力学
时空
量子场
量子引力
协变量子场


现实不似你所见
量子引力之旅
[意] 卡洛·罗韦利 著
杨光 译
引爆科普阅读热潮，现象级畅销科学读物
《七堂极简物理课》进阶版
“让物理变得性感的男人，下一个斯蒂芬·霍金”
——意大利物理学家 卡洛·罗韦利 重磅新作
从古希腊的科学发端，漫步到前沿的量子引力
延续动人的诗意与优美，凝视宇宙的奇妙本质

世界是由什么构成的

第 3 章
数据的模型与结构





Binary : 01111111 11111111 11111111 11110000
Decimal : 2147483632
Date : 2038-01-19 03:13:52 (UTC)
Date : 2038-01-19 03:13:52 (UTC)

$$\begin{array}{r} 9 \\ +\quad 1 \\ \hline 10 \end{array}$$

08:58
08:59
09:00

Year 2038 Problem

什么是进制

- 进位制是一种记数方式，亦称进位计数法或位值计数法。利用这种记数法，可以使用有限种数字符号来表示所有的数值。

基数（底数）
可使用的数字符号个数

二进制	八进制	十进制	十六进制
0	0	0	0 1 2 3
1	1	1	4 5 6 7
	2	2	8 9 A B
	3	3	C D E F
	4	4	
	5	5	
	6	6	
	7	7	
		8	
		9	

什么是进制

$$1234.56_{10} = 1\times10^{3} + 2\times10^{2} + 3\times10^{1} + 4\times10^{0} + 5\times10^{-1} + 6\times10^{-2}$$

$$10110_{2} = 1\times2^{4} + 0\times2^{3} + 1\times2^{2} + 1\times2^{1} + 0\times2^{0}$$

$$3F6B_{16} = 3\times16^{3} + 15\times16^{2} + 6\times16^{1} + 11\times16^{0}$$

位权
该位的单位值

- R 进制的位权是什么？
- 尝试对八进制数135.27进行分解

你学会了
R 进制数 → 十进制数

什么是进制

计算机为何选择二进制？

if (x<5)

True (1)

False (0)

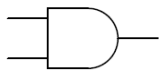
易于表示

实现简易

抗干扰

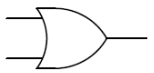
二进制在计算机中的应用

从逻辑门到集成电路



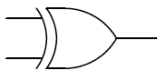
AND

A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1



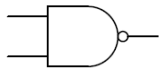
OR

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1



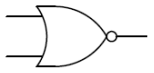
XOR

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0



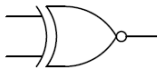
NAND

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0



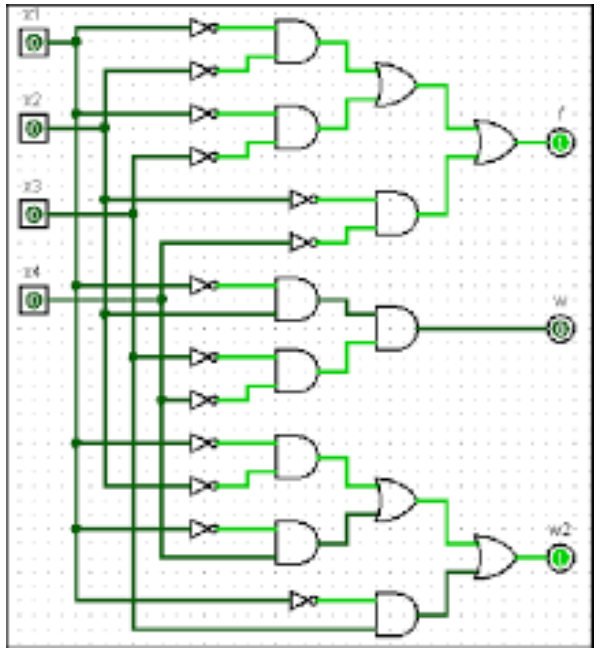
NOR

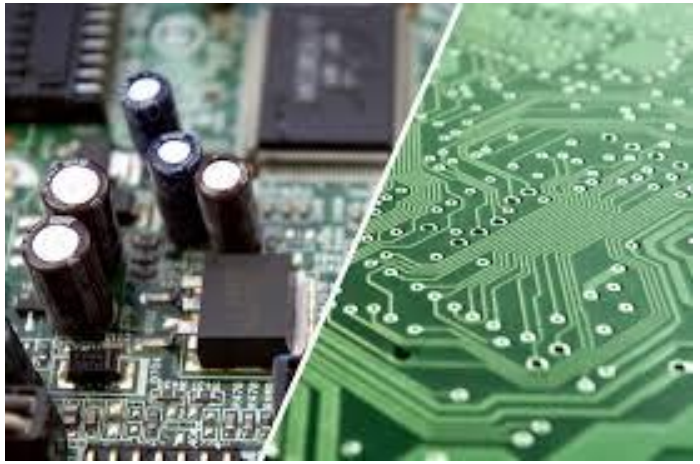
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0



XNOR

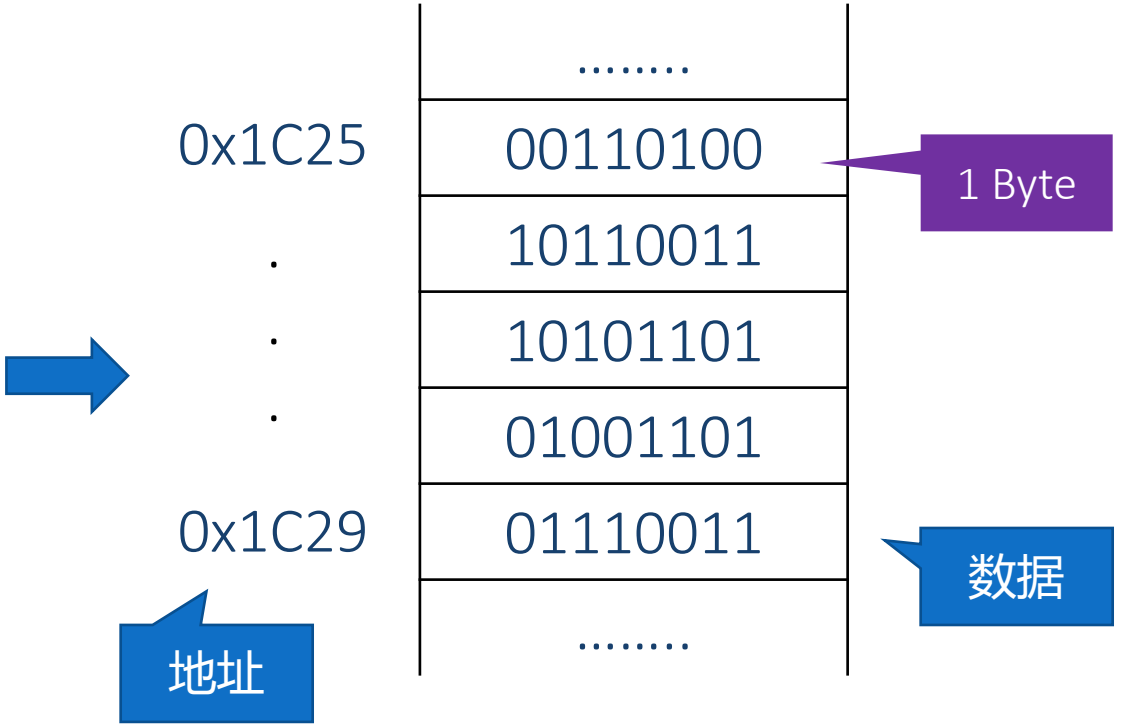
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1





二进制在计算机中的应用

第 3 章
数据的模型与结构



1 Byte = 8 bit

1 KB = 2^{10} Byte

1 MB = 2^{10} KB

G – 2^{30}

T – 2^{40}

P – 2^{50}

E – 2^{60}

二进制编码的基本概念

100Mbit/s

生产商：1 KB=1000 B
计算机：1 KB=1024 B

二进制编码的基本概念

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

- ASCII
- American Standard Code for Information Interchange
- 美国信息交换标准代码
- 基于拉丁字母的一套电脑编码系统
- 它主要用于显示现代英语。

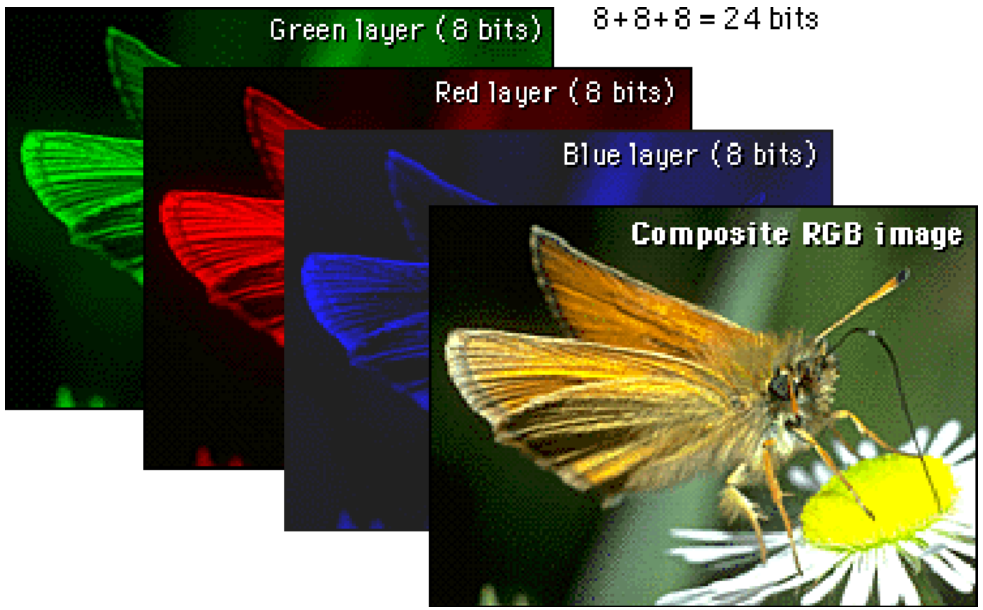
ASCII 码

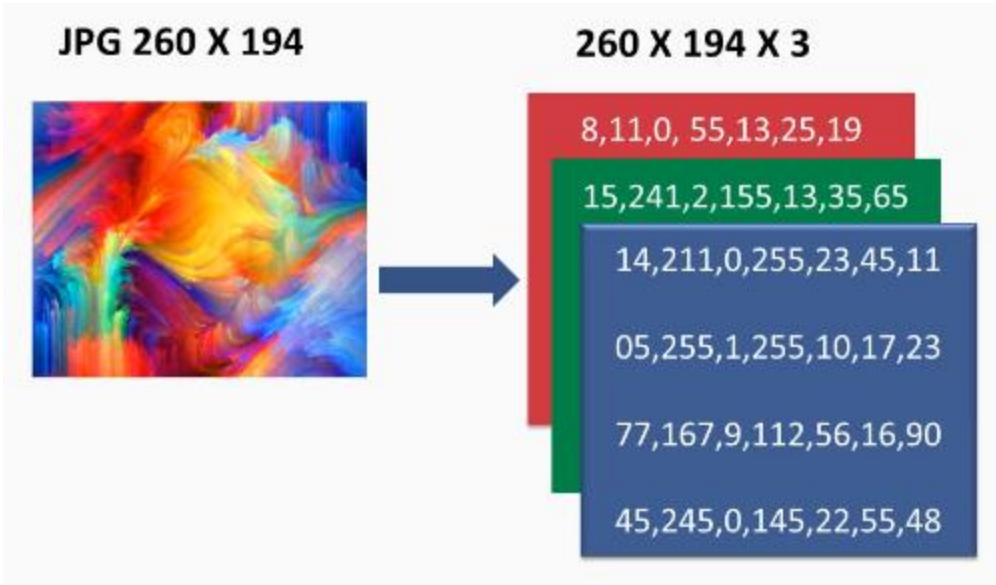
- GBK、GB 2312、GB 18030（中国内地）
- HKSCS（香港）
- Big5、CNS 11643（台湾）
- JIS（日本）
- KS X 1001（韩国）
- KPS 9566（朝鲜）

Unicode对世界上大部分的文字系统进行了整理、编码，使得电脑可以用更为简单的方式来呈现和处理文字。

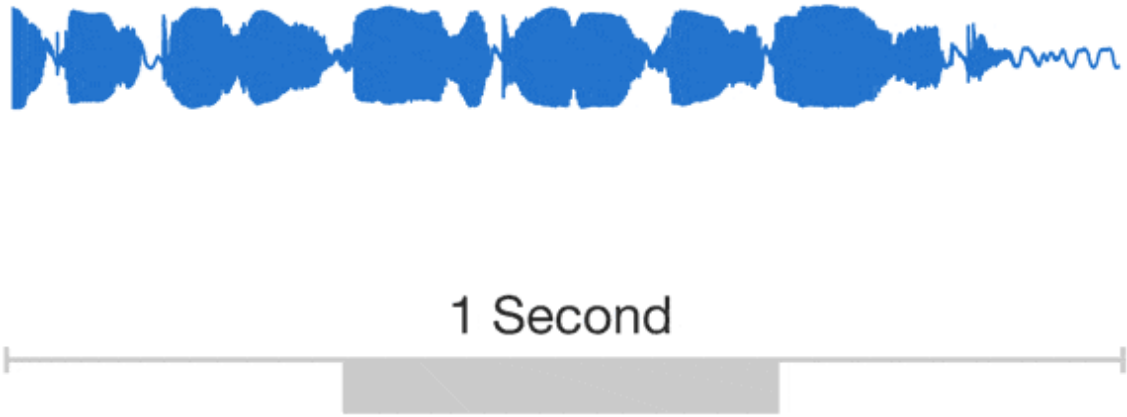
包含Emoji表情

汉字及其他语言编码


Green layer (8 bits)
8+8+8 = 24 bits
Red layer (8 bits)
Blue layer (8 bits)
Composite RGB image


JPG 260 X 194
260 X 194 X 3
8,11,0, 55,13,25,19
15,241,2,155,13,35,65
14,211,0,255,23,45,11
05,255,1,255,10,17,23
77,167,9,112,56,16,90
45,245,0,145,22,55,48

数字图像



Sound Sample

MIDI

数字音频

Frame

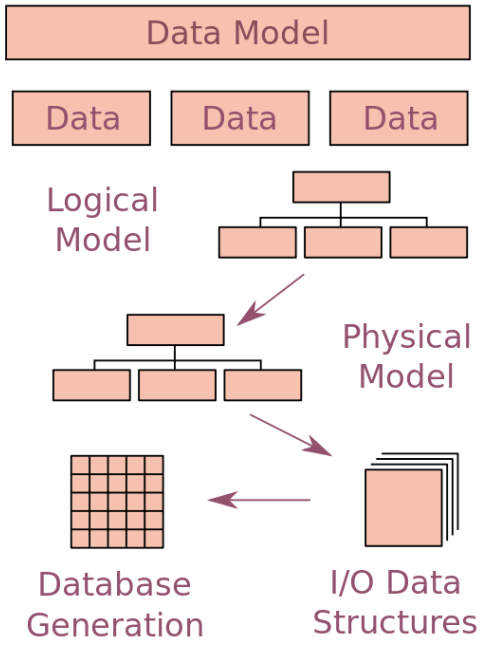
Frame per second
帧频

数字视频

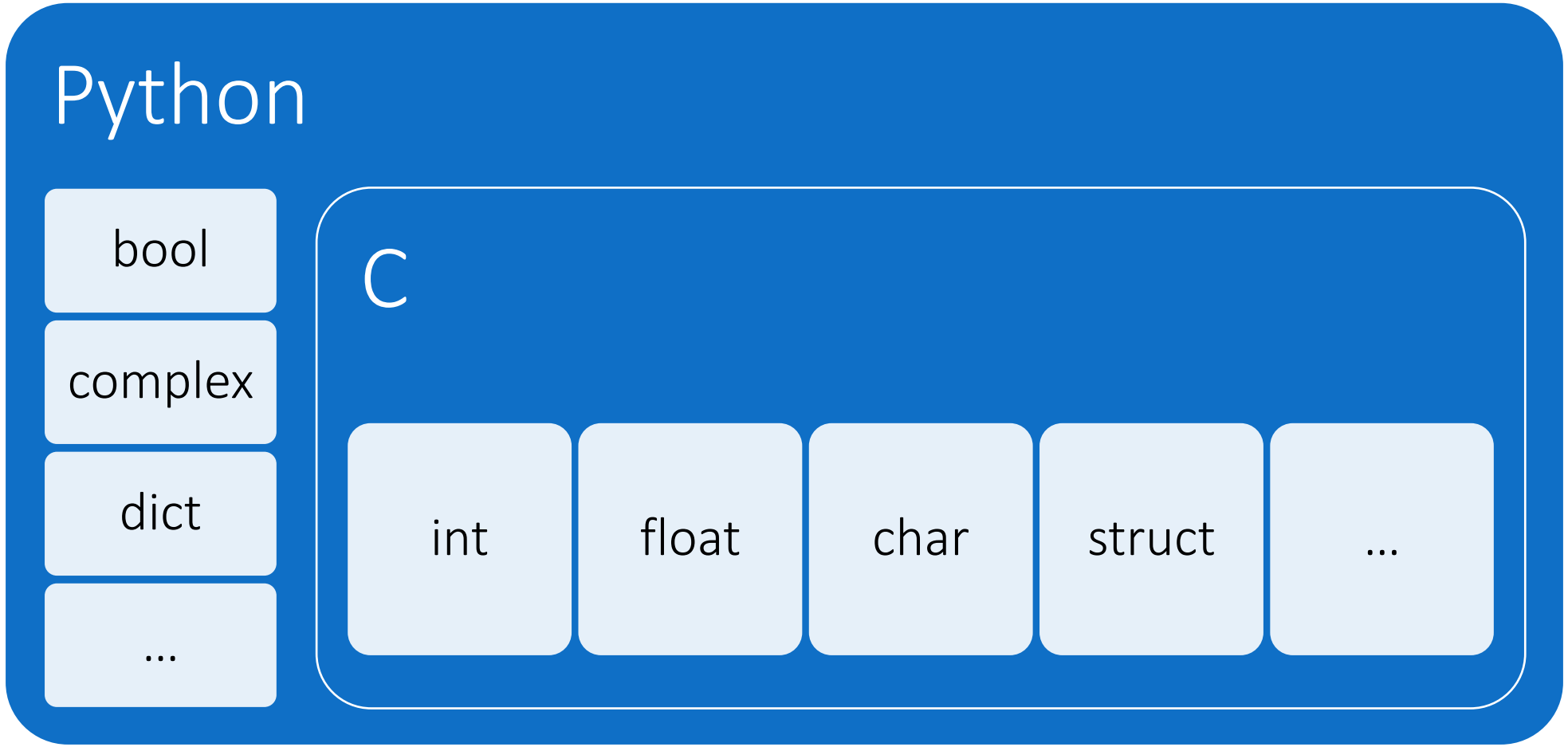
第 3 章
数据的模型与结构

- 数据模型是一种抽象模型，用于组织数据元素并标准化它们之间的关系以及与现实世界实体的属性。
- 例如，数据模型可以指定代表汽车的数据元素由许多其他元素组成，这些元素依次表示汽车的颜色、尺寸、所有者。

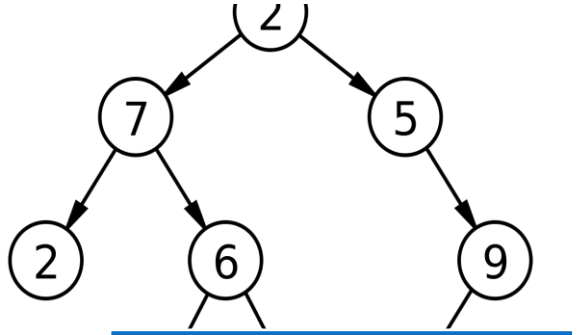
- 程序语言中的数据模型
  - 数据类型、数据结构
- 系统软件中的数据模型
  - 文件、目录、进程



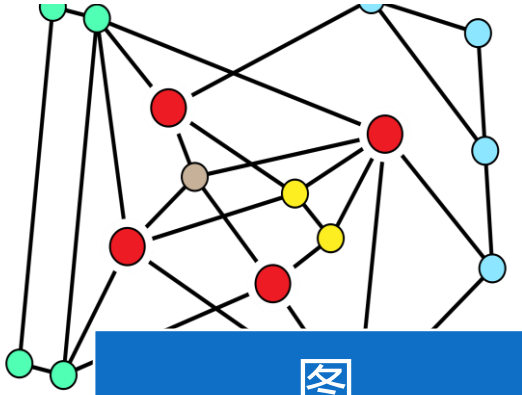
什么是数据模型



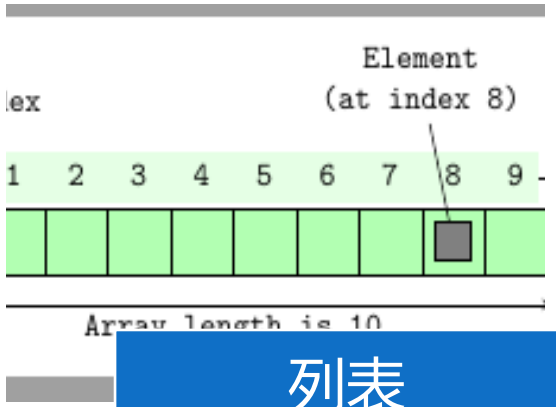
编程语言中的基本数据模型



树



图



列表

集合

编程语言中的高级数据模型

关系（Relation）数据模型

- 计算机最为重要的一项应用就是存储和管理数据。数据的组织方式对访问和管理信息的容易程度有着深刻的影响。最简单而最万能的数据组织方式就是将其存储在表中。
- 关系模型是这一概念的核心：数据被组织成称为“关系”的二维表集合。
- 关系中的每个元组都是一列，它表示每个元组中所含组分的数量。
  - 例如，表中的列都被给定了名称，称为属性（attribute）。属性分别有课程、学号和成绩。

课　程	学　号	成　绩
CS101	12345	A
CS101	67890	B
EE200	12345	C
EE200	22222	B+
CS101	33333	A−
PH100	67890	C+

编程语言中的高级数据模型

模式（Patterns）

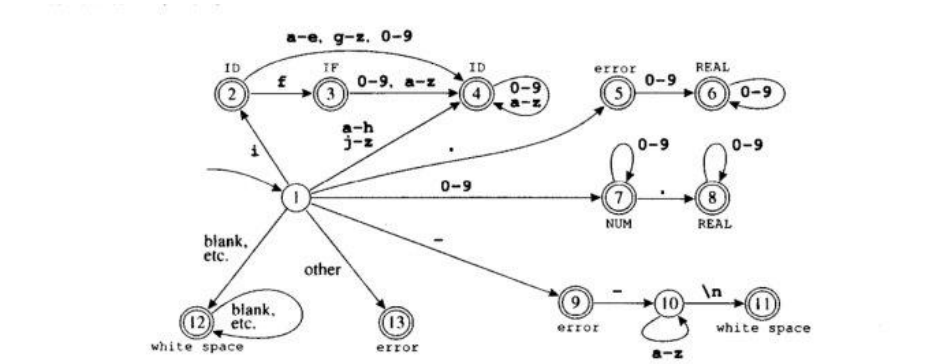
- 模式是具有某个可识别属性的对象组成的集合。字符串集合就是一类模式。
  - 比如C语言合法标识符的集合，其中每个标识符都是个字符串，由字母、数字和下划线组成，开头为字母或下划线。
  - 另一个例子是由只含0和1的给定大小数组构成的集合，读字符的函数可以将其解释为表示相同的符号。下图展示了可以解释为字母“A”的3个7×7数组。所有这样的数组就可以构成模式“A”。

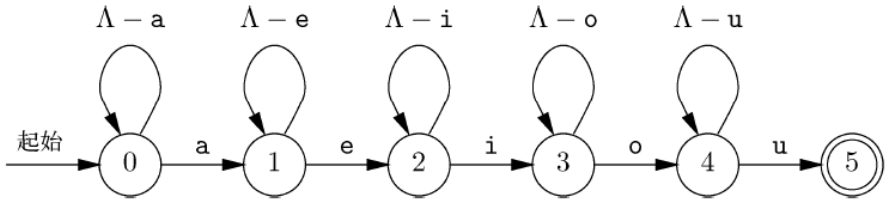
0 0 0 1 0 0 0	0 0 0 0 0 0 0	0 0 0 1 0 0 0
0 0 1 1 1 0 0	0 0 1 0 0 0 0	0 0 1 0 1 0 0
0 0 1 0 1 0 0	0 0 1 1 0 0 0	0 1 1 0 1 0 0
0 1 1 0 1 1 0	0 1 0 1 0 0 0	0 1 1 1 1 1 0
0 1 1 1 1 1 0	0 1 1 1 0 0 0	1 1 0 0 0 1 1
1 1 0 0 0 1 1	1 0 0 1 1 0 0	1 0 0 0 0 0 1
1 0 0 0 0 0 1	1 0 0 0 1 0 0	0 0 0 0 0 0 0

编程语言中的高级数据模型

自动机（Automata）

- 用来查找模式的程序通常有着特殊的结构。我们可以在代码中确定某些位置，在这些位置可以得知与程序寻找模式实例的过程有关的特殊信息。我们将这些位置称为状态。而程序的整体行为可以视作程序随着读入输入从一种状态转移到另一种状态。
- 表示程序状态的图都是有向图，它们的弧都是用字符集标记的，这样的图就被称为有限自动机，或就叫自动机。





编程语言中的高级数据模型

正则表达式（Regular Expressions）

- 自动机定义了模式，即表示自动机的图中，作为从起始状态到某个接受状态的路径标号的字符串组成的集合。
- 正则表达式与我们熟悉的算术表达式代数，以关系代数相似，可以用正则表达式代数表示的模式组成的集合，与可以用自动机描述的模式组成的集合相同。
- 正则表达式是对字符串操作的一种逻辑公式，就是用事先定义好的一些特定字符、及这些特定字符的组合，组成一个“规则字符串”，这个“规则字符串”用来表达对字符串的一种过滤逻辑。

正则表达式示例

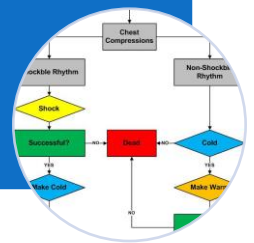
功能	正则表达式
匹配身份证号	(^\d{15}$)\|(^\d{17}([0-9]\|X)$)
匹配电子邮箱	w+([-+.]w+)@w+([-.]w+).w+([-.]w+)*
匹配手机号	\d{3}-\d{8}|\d{4}-\d{7}
中国邮政编码	d{6}
中国电话号码	((d{3,4})|d{3,4}-)?d{7,8}(-d{3})*
将一个URL解析为协议、域、端口及相对路径	/(\w+):\/\/([^/:]+)(:\d*)?([^#]*)/
匹配 HTML 标记	/<\s*(\S+)(\s[^>]*)?>[\s\S]*<\s*\/\1\s*>/

- 面向不同场景
- 丰富内置数据模型
- 接近人类世界观

对程序语言设计的指导

- 对传统算法的改进
- 突破传统数据模型限制

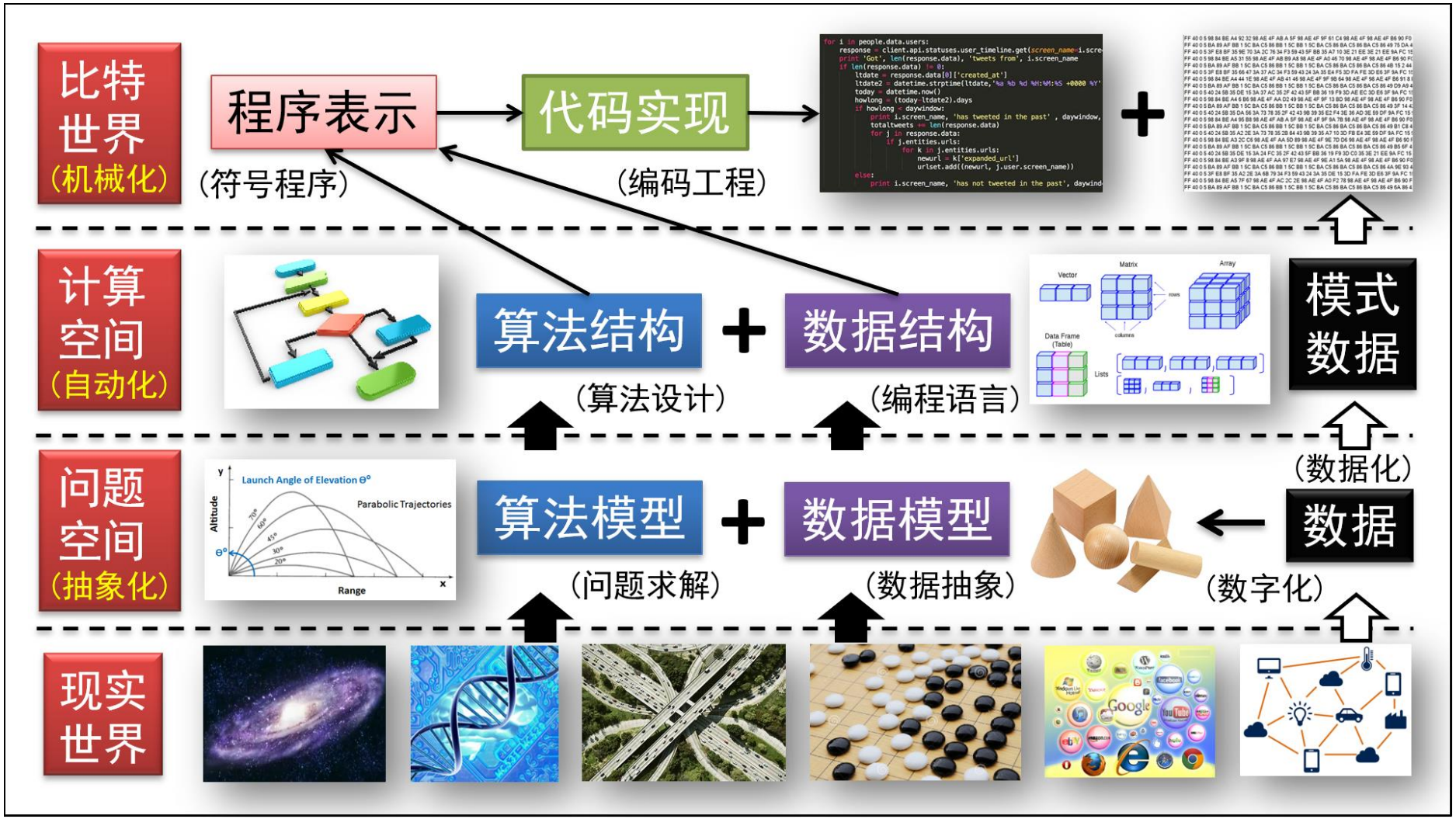
对算法设计的影响



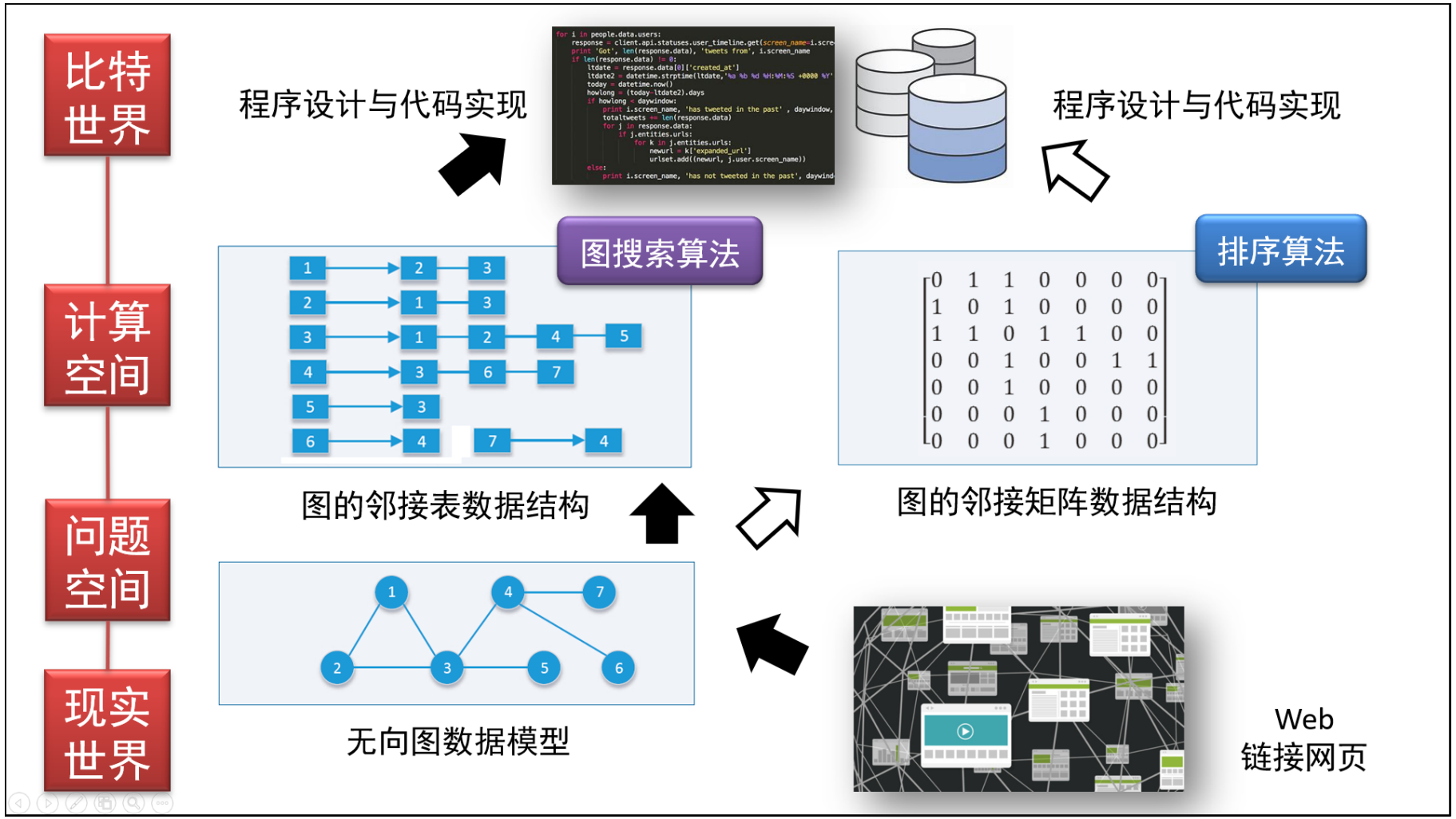
- 抽象复杂问题
- 屏蔽内部细节

对问题的构造与求解

数据模型的应用


比特世界
(机械化)
程序表示
(符号程序)
代码实现
(编码工程)
计算空间
(自动化)
算法结构
(算法设计)
数据结构
(编程语言)
模式数据
问题空间
(抽象化)
Launch Angle of Elevation θ°
Parabolic Trajectories
Altitude
Range
算法模型
(问题求解)
数据模型
(数据抽象)
(数据化)
数据
(数字化)
现实世界

数据模型的应用


比特世界
计算空间
问题空间
现实世界
程序设计与代码实现
程序设计与代码实现
图搜索算法
排序算法
图的邻接表数据结构
图的邻接矩阵数据结构
无向图数据模型
Web
链接网页

数据模型的应用

第 3 章
数据的模型与结构

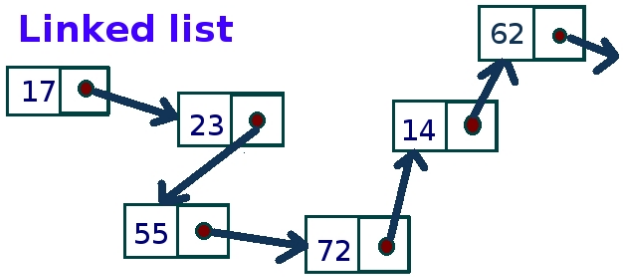
- 在数据科学中，数据结构是计算机中存储、组织数据的方式
- 数据结构意味着接口或封装
  - 一个数据结构可被视为两个函数之间的接口
  - 或者是由数据类型联合组成的存储内容的访问方法封装。
- 数据结构具体指同一类数据元素中，各元素之间的相互关系，包括三个组成部分
  - 数据的逻辑结构
  - 数据的存储结构
  - 数据的运算结构。

什么是数据结构

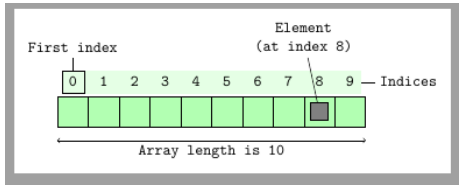
数据模型是数学抽象

数据结构是程序表达

链表



列表



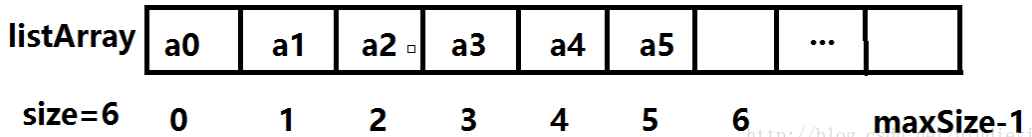
代码实现

数据结构

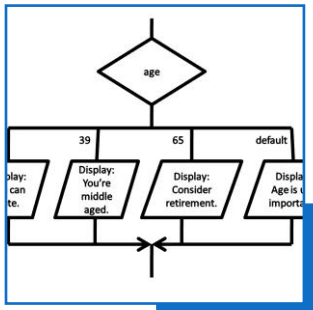
数据模型

data format

顺序表



数据模型与数据结构的关系



逻辑结构

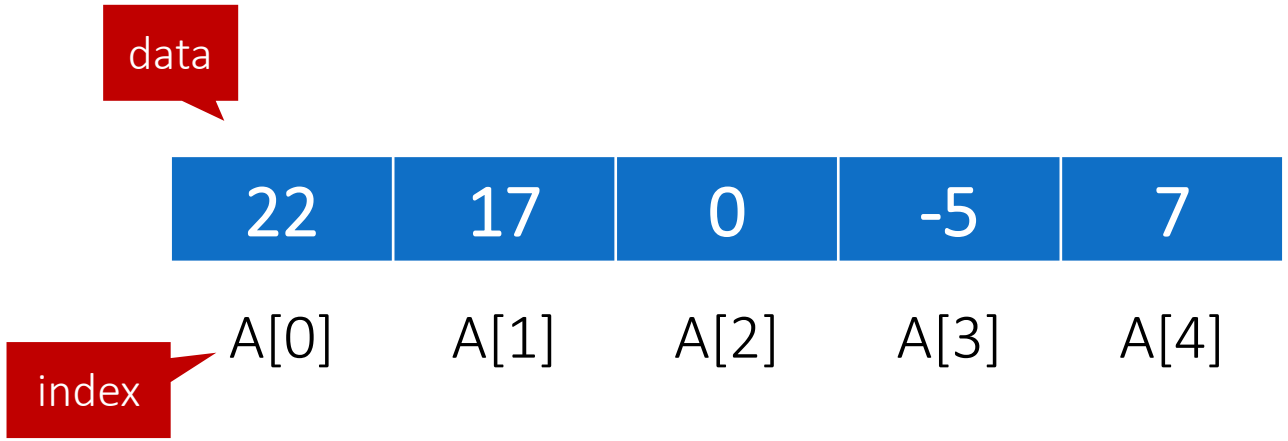
- 集合
- 线性结构
- 树形结构
- 图形结构

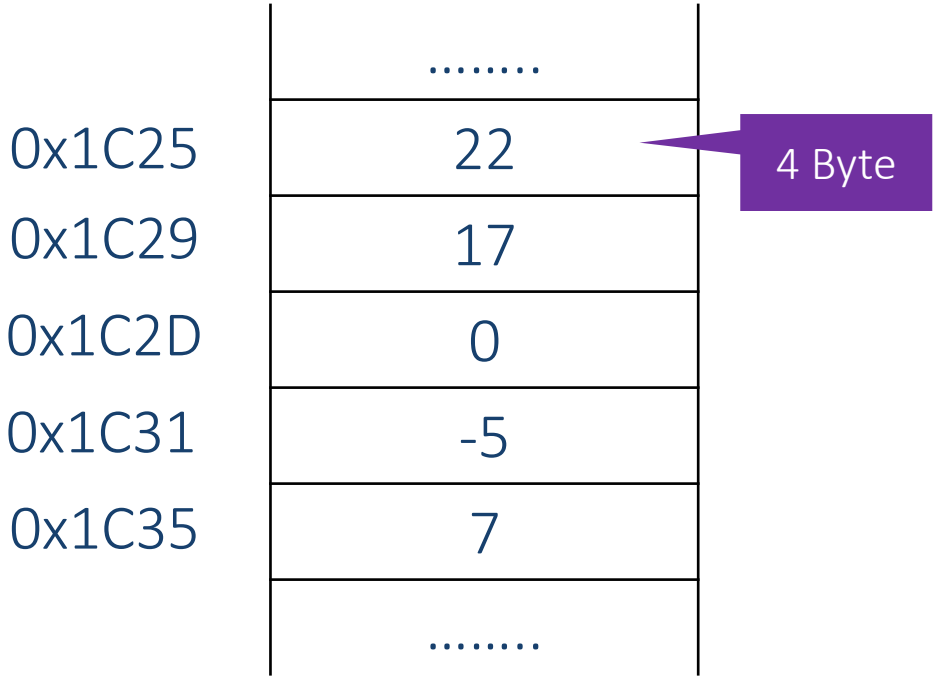
物理结构

- 数据元素的机内表示
- 关系的机内表示
  - 顺序映像
  - 非顺序映像

数据结构的研究对象

数组



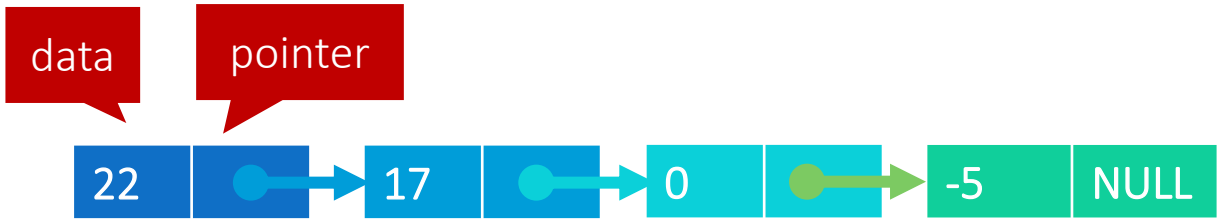


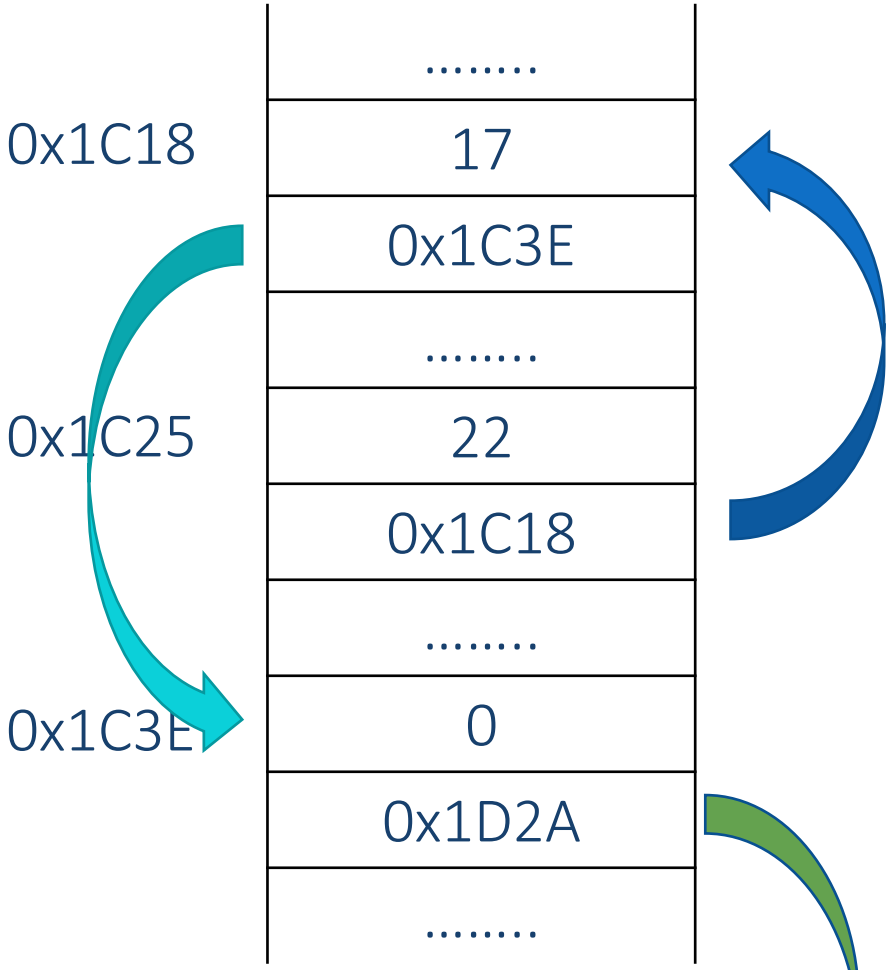
由相同类型的元素的集合所组成的数据结构，分配一块连续的内存存储。

- 整数数组
- 浮点数数组
- 字符串

常见的数据结构

链表

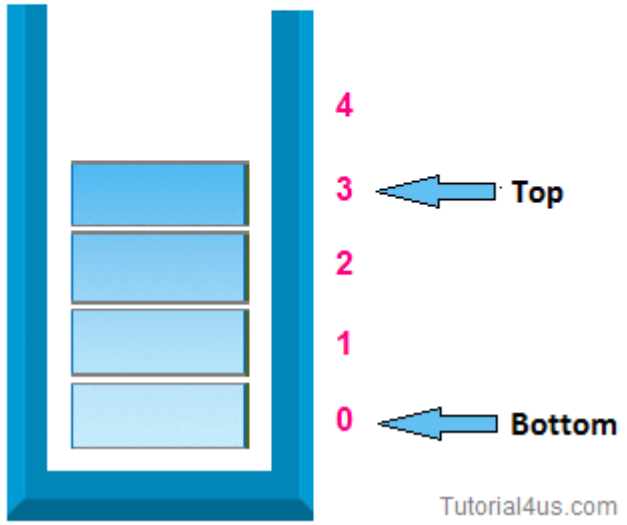




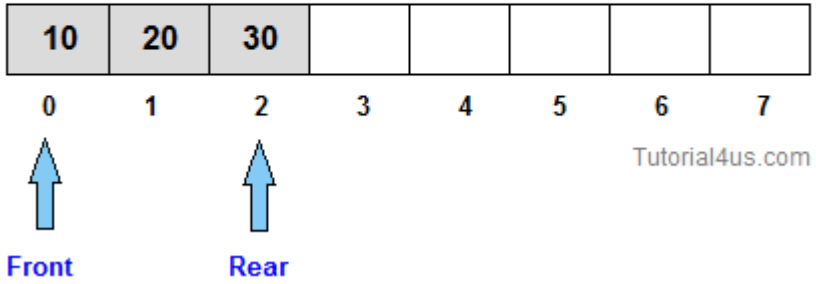
- 每个节点中存放下一节点的指针，无须按顺序存储。
- 优势
  - 插入新节点时间复杂度为常数级
  - 无需预先确定数组大小
- 劣势
  - 查找不便

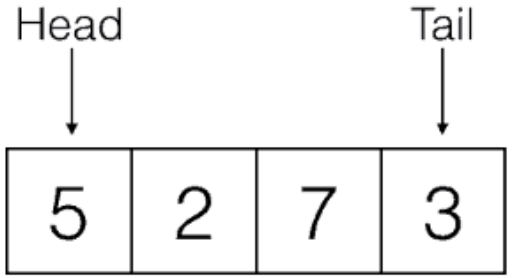
堆栈

- 只允许在有序的线性数据集合的一端（栈顶 Top）进行如下操作
  - 入栈（Push）
  - 出栈（Pop）
- 按照后进先出（LIFO）的原则运作



队列



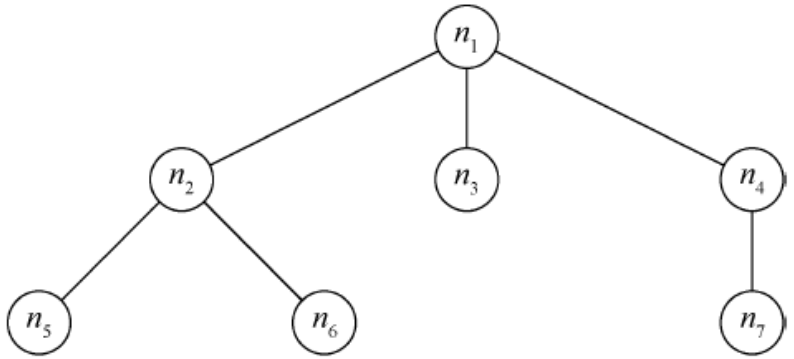


- 只允许在有序的线性数据集合的两端进行如下操作
  - 在队尾（Tail）入队
  - 在队首（Head）出队
- 按照先进先出（FIFO）的原则运作

常见的数据结构

树

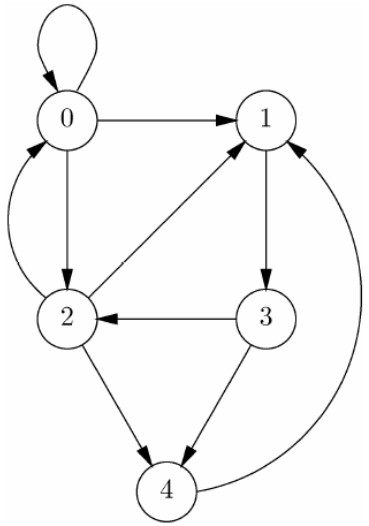
- 树是被称为节点的点与被称为边的线的集合。一条边连接着两个不同的节点，要形成树，这一系列的节点和边必须满足某些属性。
  - 在树中，有一个节点是与众不同的，它被称为根，树的根通常画在其顶端。
  - 除根之外的每个节点*c*都由一条边连接到某个称为*c*的父节点的节点*p*。我们也将节点*c*称为*p*的子节点。
  - 如果从除根之外的任一节点n开始，移动到n的父节点，再到n的父节点的父节点，以此类推，最终到达树的根节点，就说树是连通的。



图

- 从某种意义上讲，图就是二元关系。不过，它利用一系列由线（称为边）或箭头（称为弧）连接的点（称为节点）提供了强大的视觉效果。

- 图有多种形式：有向图/无向图，以及标号图/无标号图等。
- 例如，有向图，是由节点集合*N*以及*N*上的二元关系*A*组成的。我们将*A*称为有向图弧的集合，因此弧是节点的有序对。



第 3 章
数据的模型与结构