

二进制

Outline

- 导论
- 二进制数与“比特”
- 从小比特到大数字
- 从比特到字母
- 从比特到图像

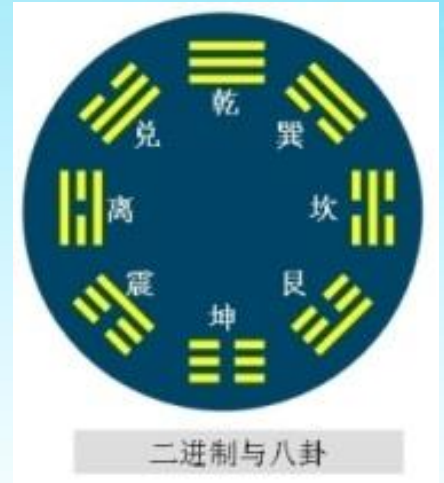
导论



- “黑客帝国”是一个二进制世界
- 计算机可以展现形式多样的信息
 - 文档、网页、照片、音乐、视频等
- 单数据无一例外是由一系列0和1的形式存储的

导论

- 二进制的起源
 - 中国《易经》以爻、卦来表示天地和万物
 - 爻分阴爻和阳爻两种
 - 德国数学家莱布尼茨受到《易经》启发在18世纪发明二进制
 - 莱布尼兹甚至在1674制作了一台“乘法机”的机械计算机



导论

- 二进制的起源

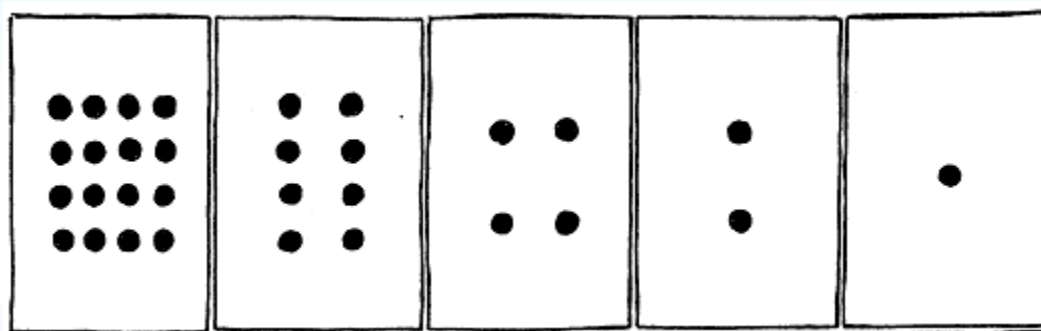
- 1与0，一切数字的神奇渊源。这是造物的秘密美妙的典范，因为，一切无非都来自上帝。——莱布尼兹
- 莱布尼兹认为认为二进制是世界上数学进制中最先进的
- 而现代计算机的运算模式也是二进制

二进制数与“比特”

- 认识二进制
 - 二进制数 (binary number)
 - 所有电子计算机中的数据都是以二进制的形式存储的
 - 它们直接影响到计算机工作的各项指标
 - 硬盘有多满？
 - 下载速度有多快？
 - 网上支付有多安全
 - 屏幕可显示的最高分辨率是多少？
 - 计算机的最快运算速度是多少？

二进制数与“比特”

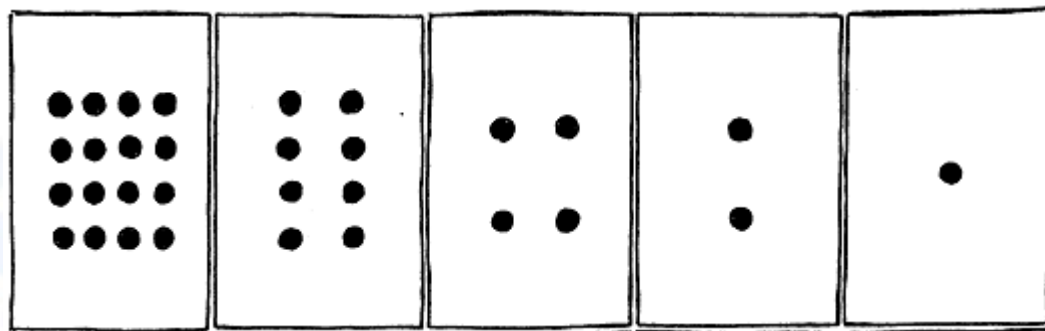
- 认识二进制
 - 二进制中仅有0和1，如何只用这两个数来数数？
 - 二进制卡片游戏



二进制数与“比特”

- 二进制卡片游戏

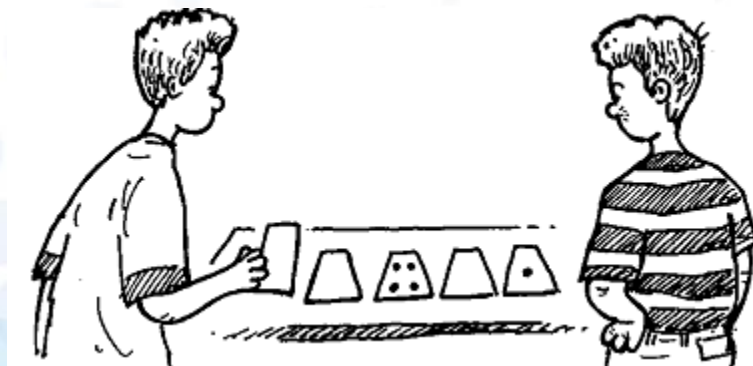
- 卡片上的圆点有怎么样的规律？
- 如果我们在这些卡片的左侧增加一张卡片，那么这张卡片上需要画几个圆点？
- 如果我们在这些卡片的左侧再增加一张卡片，那么这张新的卡片上需要画几个圆点？



二进制数与“比特”

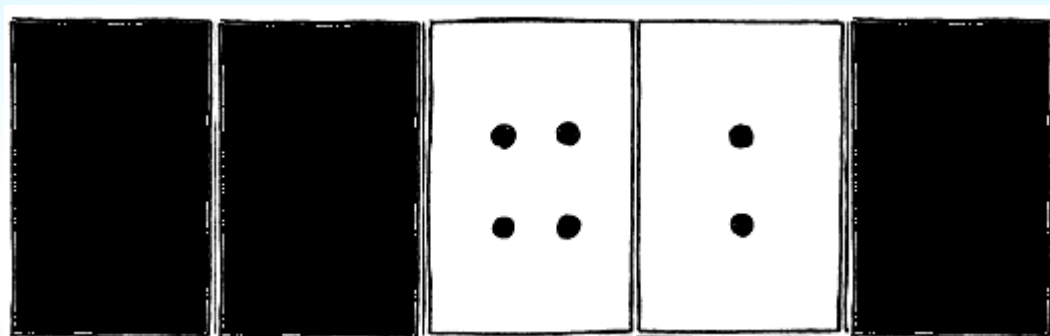
- 二进制卡片游戏

- 唯一规则：保证卡片要么正面朝上，要么翻过来背面朝上
- 将一些卡片翻转使其背面朝上，然后计算正面朝上卡片上点数之和来代表不同的数字
- 例如：下图中仅将1个圆点和4个圆点的卡片正面朝上代表了数字5



二进制数与“比特”

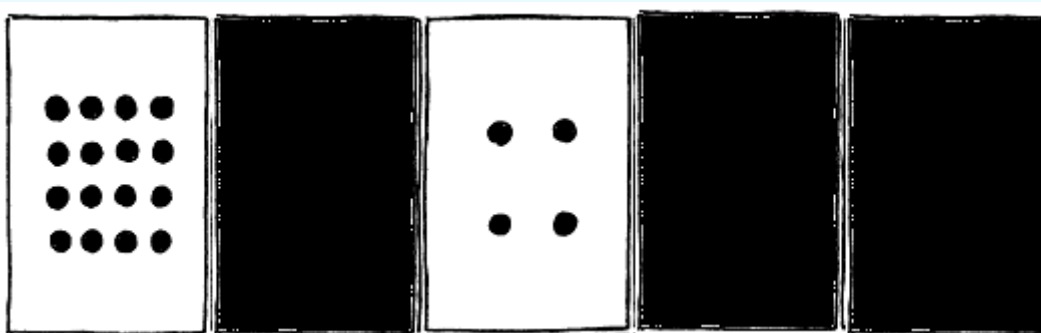
- 二进制卡片游戏
 - 你需要令那几张卡片正面朝上来表示数字6呢？



$$0 \quad 0 \quad 1 \quad 1 \quad 0 = 6$$

二进制数与“比特”

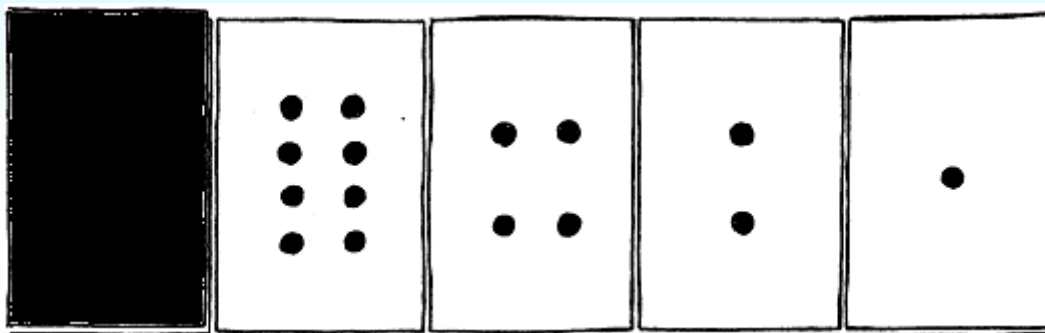
- 二进制卡片游戏
 - 你需要令那几张卡片正面朝上来表示数字20呢？



$$1 \quad 0 \quad 1 \quad 0 \quad 0 \quad = \quad 20$$

二进制数与“比特”

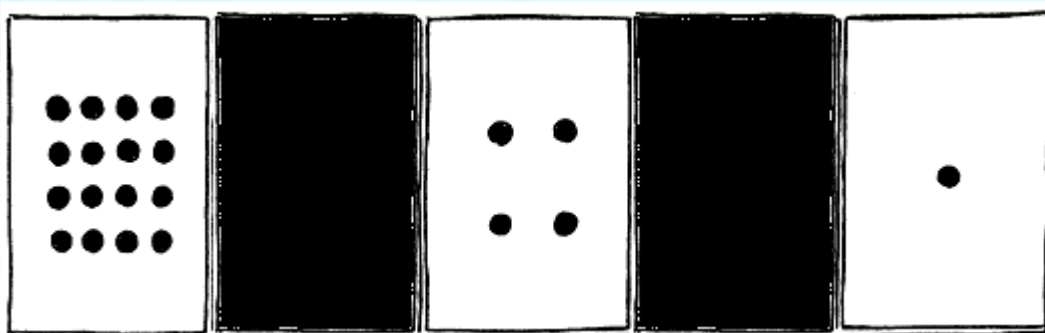
- 二进制卡片游戏
 - 你需要令那几张卡片正面朝上来表示数字15呢？



$$0 \quad 1 \quad 1 \quad 1 \quad 1 = 15$$

二进制数与“比特”

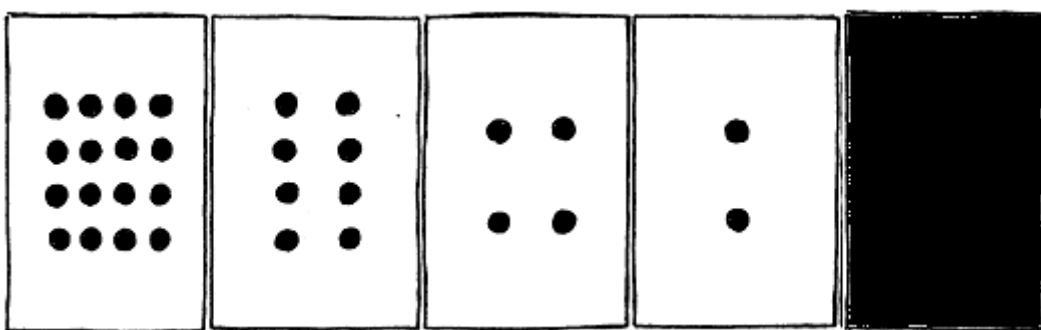
- 二进制卡片游戏
 - 你需要令那几张卡片正面朝上来表示数字21呢？



$$1 \ 0 \ 1 \ 0 \ 1 = 21$$

二进制数与“比特”

- 二进制卡片游戏
 - 你需要令那几张卡片正面朝上来表示数字30呢？



$$1 \quad 1 \quad 1 \quad 1 \quad 0 = 30$$

二进制数与“比特”

- 二进制卡片游戏

- 任何数字都能用不止一种方法来表示吗？（比如，你能用两种不同的方法来表示数字5吗？）
- 5张卡片能表示的最大数字是多少？
- 它们能表示的最小数字又是多少？
- 在0至31之间，有你无法表示的数字吗？
- 令数字递增1最简单的方法是什么？

二进制数与“比特”

- 二进制卡片游戏
 - 利用了二进制数的原理
 - 计算机仅用两个数字0和1表示信息，称之为二进制
 - 每位只有两种数值可供选择，二进制有时候也成为基数为2的数制



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

二进制数与“比特”

- 想一想

- 这些卡片一共组成多少个比特呢？
- 用卡片摆出01101，它对应的十进制数为多少？
- 二进制00110对应的十进制是多少？
- 二进制01110对应的十进制是多少？
- 二进制10001对应的十进制是多少？
- 用6个卡片最大能表示多少？



从小比特到大数字

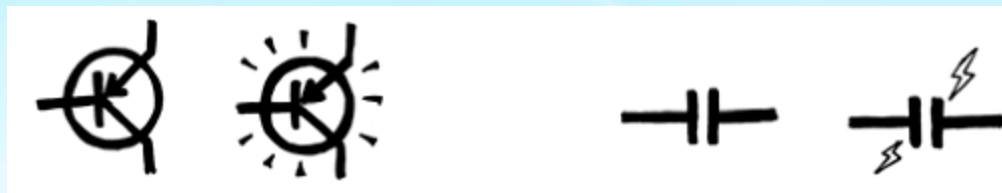
- 你一定遇到过这些术语
 - 24位色彩
 - 100兆连接速度
 - 32位计算机
 - 128位SSL加密



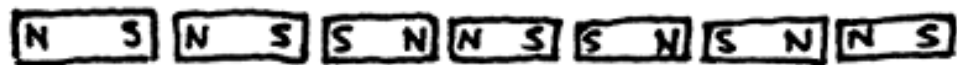
从小比特到大数字

- 计算机如何表示二进制

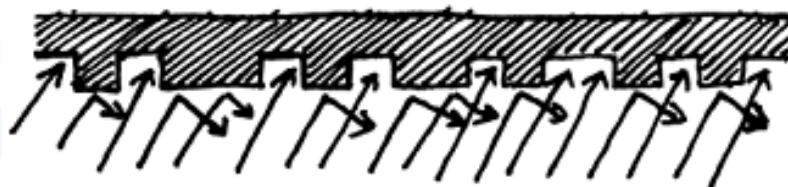
- 利用晶体管或电容表示二进制（如内存）



- 利用磁盘表面的磁介质状态表示（硬盘、磁带）

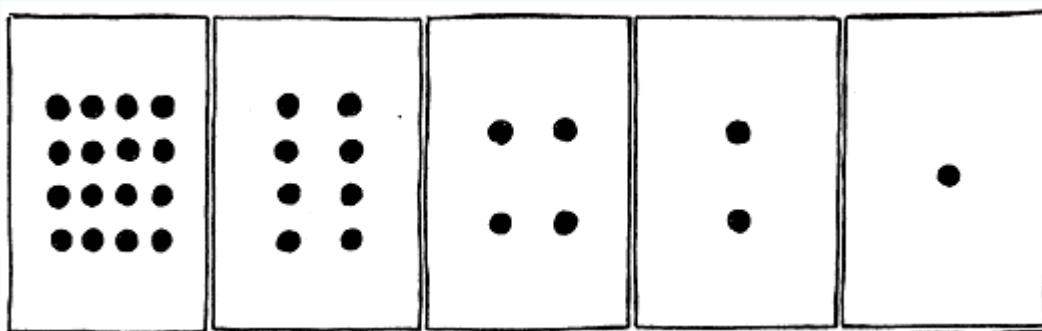


- 利用碟片表面反光特性表示（CD或DVD等）



从小比特到大数字

- 二进制数的性质
 - 卡片上的数字有什么规律？
 - 写出来是 1, 2, 4, 8, 16 ...



从小比特到大数字

- 二进制数的性质

- 把前两张卡片的数字相加($1+2$)，其计算结果是多少？
- 把前三张卡片的数字相加($1+2+4$)，其计算结果是多少？
- 那么 $1+2+4+8$ 的计算结果呢？
- 最后把五张卡片的数字全部加在一起($1+2+4+8+16$)，其结果等于多少？
- 现在你发现前面卡片的数字之和与后面一张卡片的关系了吗？

从小比特到大数字

- 二进制数的性质

- 在十进制系统中，当往数字右侧插入一个0时，就相当于将数字扩大10倍

- $9 \rightarrow 90$

- $30 \rightarrow 300$

- 当你向二进制的右侧添加一个0呢？

- $010 \rightarrow 0100$

- $101 \rightarrow 1010$

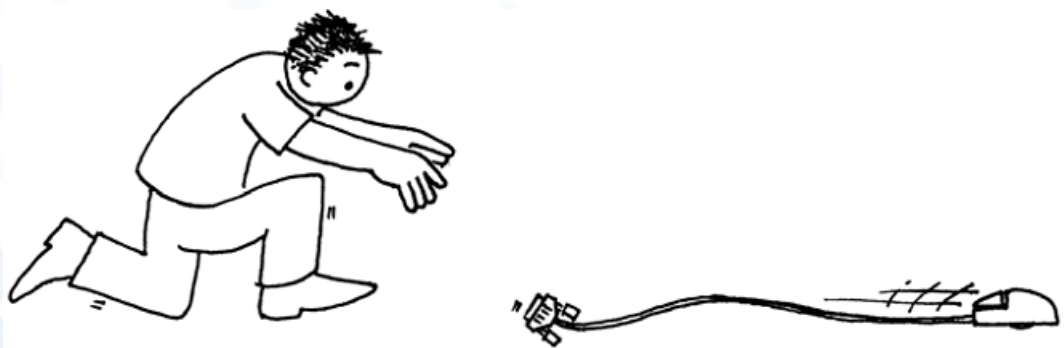
- $110 \rightarrow 1100$



从小比特到大数字

- 二进制数的性质

- 二进制数010右侧插入0后会变成什么数字？
- 二进制数101右侧插入0后会变成什么数字？
- 二进制数110右侧插入0后会变成什么数字？
- 二进制数右侧插入0后会变成一个新数字，计算该新数的规律是什么？



从小比特到大数字

- 大一点的二进制数
 - 使用5比特最大可以表示31
 - 使用6比特最大可以表示多少？
 - 使用7比特最大可以表示多少？
 - 那么8比特呢？
 - 16比特或24比特呢？如何计算？

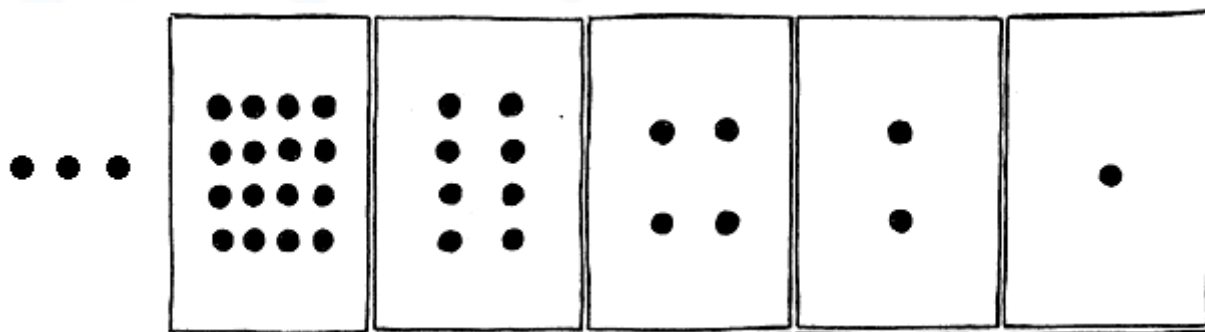


在实际应用中，计算机系统处理连续存储的8位数据时非常便利的，一般将上述连续存储的8比特称为一个字节 (byte)

从小比特到大数字

- 如何转换十进制到二进制
 - 假设我们使用8比特表示数值
 - 试试用二进制数怎样得到77
 - 写下下面的数字

128 64 32 16 8 4 2 1



从小比特到大数字

- 如何转换77到二进制
 - 从128开始
 - 这个数太大了，无论如何都无法表示出77来
 - 因此划掉

~~128~~ 64 32 16 8 4 2 1

从小比特到大数字

- 如何转换77到二进制
 - 接下来是64
 - 这个数比77小，保留，余下的数为 $77-64=13$
 - 保留

~~128~~ 64 32 16 8 4 2 1

从小比特到大数字

- 如何转换77到二进制
 - 接下来是32
 - 这个数比余下的数13要小
 - 因此划掉

~~128~~ 64 ~~32~~ 16 8 4 2 1

从小比特到大数字

- 如何转换77到二进制
 - 以此类推
 - 把划掉的数字用0表示，保留的数字用1表示
 - 得到下面的二进制数

128	64	32	16	8	4	2	1
0	1	0	0	1	1	0	1

从小比特到大数字

- 如何转换

- 如何用二进制数来表示165？
- 如何用二进制数来表示99？
- 如何用二进制数来表示127？
- 如何用二进制数来表示1000？
- 如何用二进制数来表示1024？



165→10100101 99→1100011 127→1111111
1000→1111101000 1024→10000000000

从小比特到大数字

- 术语理解

- “24位色彩”

- 是指一张图片在计算机中可用高达 $2^{24} = 16777216$ 个不同色彩元来表示或存储
 - 这是相当高清晰的图画，其实人眼是无法识别这么多的色彩元

- “100兆连接”

- 是指计算机和网络之间的连接速度能达到每秒100兆比特

从小比特到大数字

- 术语理解

- “32位计算机”

- 是指一次性储存或运算32比特的计算机
 - 处理相同的数据，一台8位计算机至少需要耗费相当于32位计算机4倍的时间才能完成

- “128位SSL加密”

- 是指用128位密钥为数据加密
 - 如果你设定了8位密码，意味着别人仅尝试256次便能入侵你的系统
 - 而128位则需尝试 $2^{128} = 340282366920938463463374607431768211456$ 次

从小比特到大数字

• 读心术

32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63

16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63

8	9	10	11
12	13	14	15
24	25	26	27
28	29	30	31
40	41	42	43
44	45	46	47
56	57	58	59
60	61	62	63

4	5	6	7
12	13	14	15
20	21	22	23
28	29	30	31
36	37	38	39
44	45	46	47
52	53	54	55
60	61	62	63

2	3	6	7
10	11	14	15
18	19	22	23
26	27	30	31
34	35	38	39
42	43	46	47
50	51	54	55
58	59	62	63

1	3	5	7
9	11	13	15
17	19	21	23
25	27	29	31
33	35	37	39
41	43	45	47
49	51	53	55
57	59	61	63



从小比特到大数字

• 读心术

– 将包含某个数字的卡片标上1，其他为0

第一个 数的值	2^5	2^4	2^3	2^2	2^1	2^0
32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63	8 9 10 11 12 13 14 15 24 25 26 27 28 29 30 31 40 41 42 43 44 45 46 47 56 57 58 59 60 61 62 63	4 5 6 7 12 13 14 15 20 21 22 23 28 29 30 31 36 37 38 39 44 45 46 47 52 53 54 55 60 61 62 63	2 3 6 7 10 11 14 15 18 19 22 23 26 27 30 31 34 35 38 39 42 43 46 47 50 51 54 55 58 59 62 63	1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63	

17 =	0	1	0	0	0	1
63 =	1	1	1	1	1	1
23 =	0	1	0	1	1	1

从小比特到大数字

- 负数在计算机中如何表示呢？
 - 在8位机中+8表示为00001000，那么-8呢？
 - 将一个二进制位规定为符号位，它等于0时就表示正数，等于1时就表示负数
 - 直观上，-8就是10001000
 - 但是计算机内部采用2的补码表示负数

从小比特到大数字

- 什么是2的补码？

- 这是一种数值转换方法，分为两步

1. 每一个二进制为都取相反值，0变1，1变0
2. 将上一步得到的值加1

- 例如求0001000的补码：

1. 00001000 $\rightarrow$ 11110111
2. 11110111 $\rightarrow$ 11111000
3. 因此-8在计算机中就是用11111000表示



为什么使用补码来表示负数呢？

从小比特到大数字

- 为什么使用补码？
 - 2的补码的便利之处体现在，所有的加法可以使用同一种电路完成

计算16+(-8)	
直觉表示法	2的补码表示法
$\begin{array}{r} 00010000 \\ + 10001000 \\ \hline 10011000 \end{array}$	$\begin{array}{r} 00010000 \\ + 11111000 \\ \hline 00001000 \end{array}$
结果为-24	结果为8

从比特到字母

- 比特表达字母与文字
 - 在计算机中看到或输入的任何文本都是0和1表达的
 - 任何与计算机连接的存储设备，硬盘、光驱、移动设备都是二进制系统
 - 计算机间互传的文档、电子邮件以及浏览的网页甚至网络上所有的一切，包括图片、音乐和视频，同样是采用二进制数表达的
 - 计算机如何用比特表达字母和文字的呢？

从比特到字母

- 一种可能的表达

- 为了表示汉字和拼音系统，就需要4个声调和26个字母，一共30个元素

1	2	3	4	5	6	7	8	9	10
—	/	∨	∖	a	b	c	d	e	f
11	12	13	14	15	16	17	18	19	20
g	h	i	j	k	l	m	n	o	p
21	22	23	24	25	26	27	28	29	30
q	r	s	t	u	v	w	x	y	z

- 可以用下面的代码表达出“你好”

18	13	3	0	12	5	19	3
n	i	3		h	a	o	3

从比特到字母

- 编码和解码

- 用预定的方法将供人们阅读的信息转换为计算机可以读取的信息称为“编码”
- 例如上例中，将“你好”改写为代码“18 13 3 0 12 5 19 3”就是编码过程
- 而将二进制信息转换为可供人们阅读的信息的过程称为“解码”
- 上例中，将代码“18 13 3 0 12 5 19 3”显示为“你好”就是解码过程

从比特到字母

- 问题

- 使用上表中的十进制编码计算机如何表示字母 d ?
- 如何表示字母y呢 ?
- 上表中数字30代表的字母是什么 ?
- 使用上表中的十进制编码如何表示单词“hello” ?
- 对数字序列“17 5 3”进行解码能得到什么 ?
- 采用上述编码方式，你的名字如何表示呢 ?

从比特到字母

- 真实的编码
 - ASCII码

	Cod.	Char	Byte	Cod.	Char
0000	64	@	01100000	96	`
0001	65	A	01100001	97	a
0010	66	B	01100010	98	b
0011	67	C	01100011	99	c
0100	68	D	01100100	100	d
0101	69	E	01100101	101	e
0110	70	F	01100110	102	f
0111	71	G	01100111	103	g
1000	72	H	01101000	104	h
1001	73	I	01101001	105	i
1010	74	J	01101010	106	j
1011	75	K	01101011	107	k
1100	76	L	01101100	108	l

Ctrl	Dec	Hex	Char	Code	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
^@	0	00		NUL	32	20	!	64	40	@	96	60	'
^A	1	01		SOH	33	21	!	65	41	A	97	61	a
^B	2	02		STX	34	22	"	66	42	B	98	62	b
^C	3	03		ETX	35	23	#	67	43	C	99	63	c
^D	4	04		EOT	36	24	\$	68	44	D	100	64	d
^E	5	05		ENQ	37	25	%	69	45	E	101	65	e
^F	6	06		ACK	38	26	&	70	46	F	102	66	f
^G	7	07		BEL	39	27	,	71	47	G	103	67	g
^H	8	08		BS	40	28	(	72	48	H	104	68	h
^I	9	09		HT	41	29	)	73	49	I	105	69	i
^J	10	0A		LF	42	2A	*	74	4A	J	106	6A	j
^K	11	0B		VT	43	2B	+	75	4B	K	107	6B	k
^L	12	0C		FF	44	2C	,	76	4C	L	108	6C	l
^M	13	0D		CR	45	2D	-	77	4D	M	109	6D	m
^N	14	0E		SO	46	2E	.	78	4E	N	110	6E	n
^O	15	0F		SI	47	2F	/	79	4F	O	111	6F	o
^P	16	10		DLE	48	30	0	80	50	P	112	70	p
^Q	17	11		DC1	49	31	1	81	51	Q	113	71	q
^R	18	12		DC2	50	32	2	82	52	R	114	72	r
^S	19	13		DC3	51	33	3	83	53	S	115	73	s
^T	20	14		DC4	52	34	4	84	54	T	116	74	t
^U	21	15		NAK	53	35	5	85	55	U	117	75	u
^V	22	16		SYN	54	36	6	86	56	V	118	76	v
^W	23	17		ETB	55	37	7	87	57	W	119	77	w
^X	24	18		CAN	56	38	8	88	58	X	120	78	x
^Y	25	19		EM	57	39	9	89	59	Y	121	79	y
^Z	26	1A		SUB	58	3A	:	90	5A	Z	122	7A	z
^[	27	1B		ESC	59	3B	;	91	5B	[	123	7B	{
^\	28	1C		FS	60	3C	<	92	5C	\	124	7C	
^]	29	1D		GS	61	3D	=	93	5D	]	125	7D	}
^^	30	1E	▲	RS	62	3E	>	94	5E	^	126	7E	~
^-	31	1F	▼	US	63	3F	?	95	5F	_	127	7F	␣*

* ASCII code 127 has the code DEL. Under MS-DOS, this code has the same effect as ASCII 8 (BS). The DEL code can be generated by the CTRL + BKSP key.

从比特到字母

- 真实的编码
 - ASCII中每个字符用7比特来表示
 - 因此一共可以表示128个字符，这对于英语已经足够
 - 对于拥有更大字符容量的字库，比如中文，计算机使用Unicode系统，其中每个字符用16比特来表示



用16比特来编码一共能表示多少个不同的字符？

从比特到字母

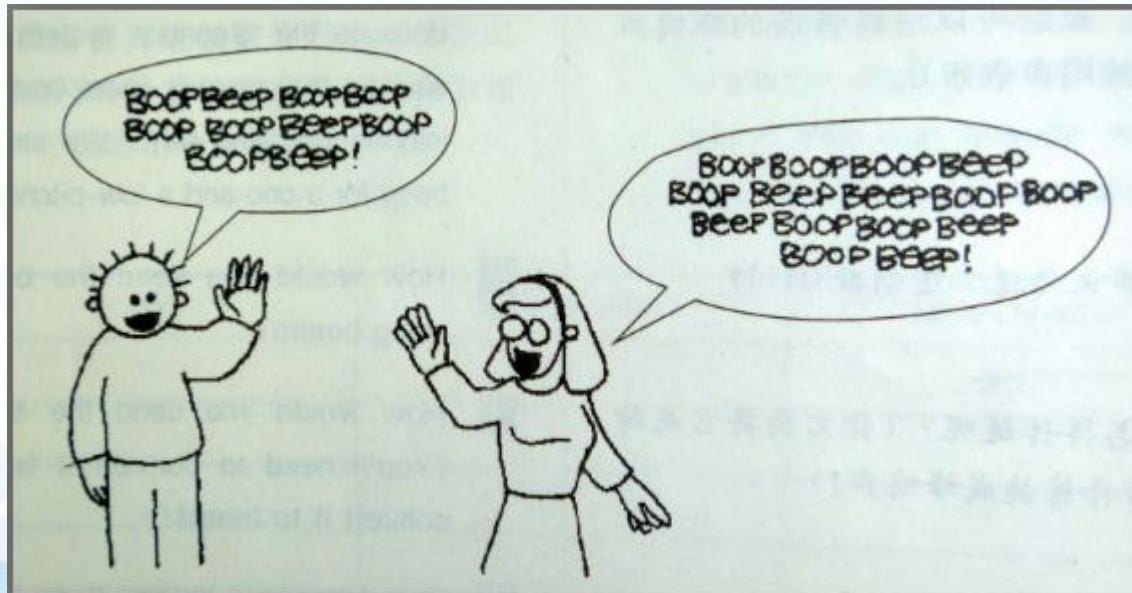
- 传音游戏

- 二进制系统可以用来表示计算机中所有的数据，所以当计算机通过调制解调器连接到网络时也是用二进制发送信息的
- 电话系统是用来承载信号的，它使用蜂鸣声来表达传送的声音信息
- 我们可以用高声调的蜂鸣声表示1，低音调的蜂鸣声表示0

从比特到字母

- 传音游戏

- 那么怎么用蜂鸣声来传送二进制0111？
- 十进制9用怎样传送呢？
- 如何用字母传送字母“q”？



从比特到字母

- 莫尔斯电码
 - 莫尔斯电码的原理也是类似的
 - 它有两种“符号”用来表示字符点(.)和划(-)或称为“滴”(Dit)和“答”(Dah)



A	• —	U	• • —
B	• • • —	V	• • • —
C	— • • —	W	• — • —
D	— • •	X	— • • —
E	•	Y	— • — —
F	• • — •	Z	— — • •
G	• — — •		
H	• • • •		
I	• •		
J	• — — —		
K	— • — —		
L	• — • •		
M	— — •		
N	— • —		
O	— — —		
P	• — — •		
Q	— • — —		
R	• — • •		
S	• • •		
T	—		
		1	• — — — —
		2	• • — — —
		3	• • • — —
		4	• • • • —
		5	• • • • •
		6	— • • • •
		7	— — • • •
		8	— — — • •
		9	— — — — •
		0	— — — — —

从比特到字母

- 莫尔斯电码

- SOS ...——...
- 外星人 ET .—
- TVB —...——...



香港的TVB自开台1967年起，每次新闻报道在背景音乐中播放的“NEWS TODAY”音频。这段背景音乐就是“TVB”的摩尔斯电码的声音。



从比特到字母

- 《风声》中莫尔斯电码



利用线缝制的莫尔斯电码



利用唱歌曲调哼出的莫尔斯电码

从比特到图像

- 如何显示图片

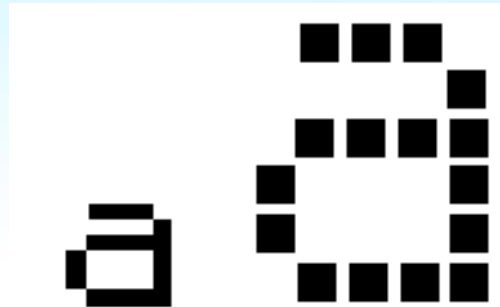
- 一张图片看起来细腻光滑
- 当我们放大它眼睛的部分，却发现细节过渡并没有那么平滑
- 当我们将图片放得更大时，你会注意到，图片是由一大堆小方块拼成的，他们被称之为像素



从比特到图像

- 图像表示

- 在简单的黑白图像中，每个像素只有两种值：
黑或白



- 上图是字母“a”的放大图
- 计算机存储这样的图片时，它只需要记录图像中那些是黑点那些是白点

从比特到图像

- 存储字母“a”的简单方式
 - 每个像素用一个比特代替
 - 如果是白色像素，存为0
 - 如果是黑色像素，存为1

	■	■	■		0, 1, 1, 1, 0
				■	0, 0, 0, 0, 1
	■	■	■	■	?, ?, ?, ?, ?
■				■	1, 0, 0, 0, 1
■				■	?, ?, ?, ?, ?
	■	■	■	■	?, ?, ?, ?, ?

从比特到图像

- 图像压缩

- 同样的方法可以存储更复杂的黑白图像
- 但是存储大图片时需要用到许多比特，因为每个像素必须使用一个比特表示
- 这样浪费更大的空间和更多的时间
- 于是人们发明了许多不同的方法来减少储存或发送图片需要占用的体积，减小文件体积的过程被称为压缩

从比特到图像

- 游程压缩

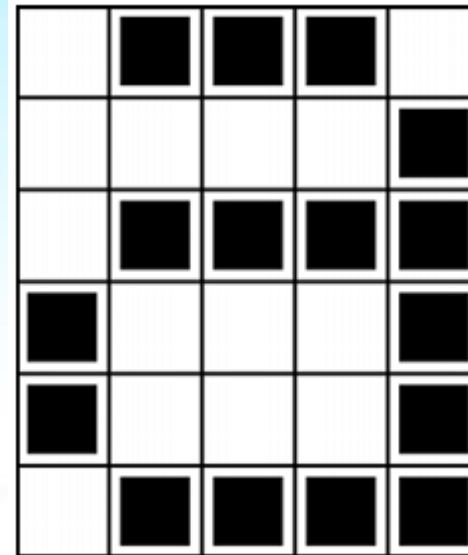
- 因为在图片中会有大块连续的白色像素以及连续的黑色像素的特点
- 因此只须记录下每个白色或黑色像素连续区块的长度，就可以表示整张图像
- 这种方式被称为游程压缩或游程编码(RLE)



因为这种方法很适合用来发送文本页面，因此传真机也是采用的同样的原理

从比特到图像

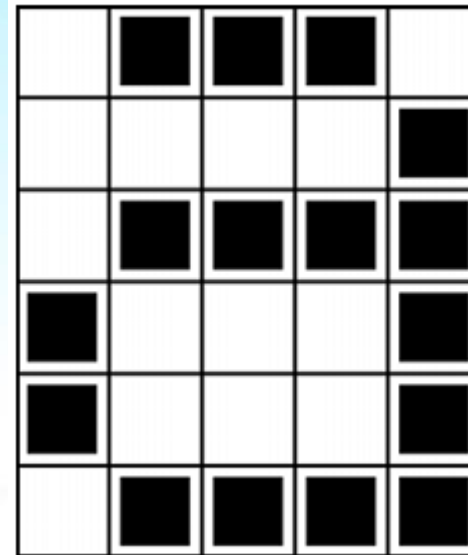
- 字母“a”的例子
 - 第一行包含1个白色像素，接着3个黑色像素，接着1个白色像素
 - 因此第一行会被编码为1, 3, 1



1, 3, 1

从比特到图像

- 字母“a”的例子
 - 第二行包含4个白色像素，接着1个黑色像素
 - 因此第二行会被编码为4, 1
 - 计算机如何编码第3行呢？



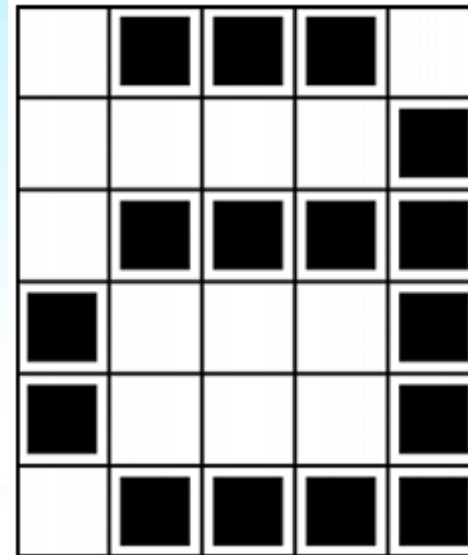
1, 3, 1

4, 1

1, 4

从比特到图像

- 字母“a”的例子
 - 但是如何编码第4行呢？
 - 因为第4行并不是以白色像素打头的，因此在开头加入一个0
 - 编码为0, 1, 3, 1
 - 其他行如何编码呢？



1, 3, 1

4, 1

1, 4

0, 1, 3, 1

0, 1, 3, 1

1, 4

从比特到图像

- 游程编码的问题

- 因为游程的长度在计算机中也是采用二进制表示的
- 因此能够表达出的游程长度值存在一个上限
- 因此能够表达的长度值存在一个上限



例如，如果计算机中设定用5比特表达游程长度，那么这个长度值就无法超过31

从比特到图像

- 游程编码的问题

- 如果可用的最大值为31，那么怎样来表示一段长为36的连续黑色像素呢？



可以将这个连续的黑色像素段拆成两端，那么就是两段长度分别是31和5的像素段，编码应该为 0, 31, 0, 5

从比特到图像

- 传真机如何工作

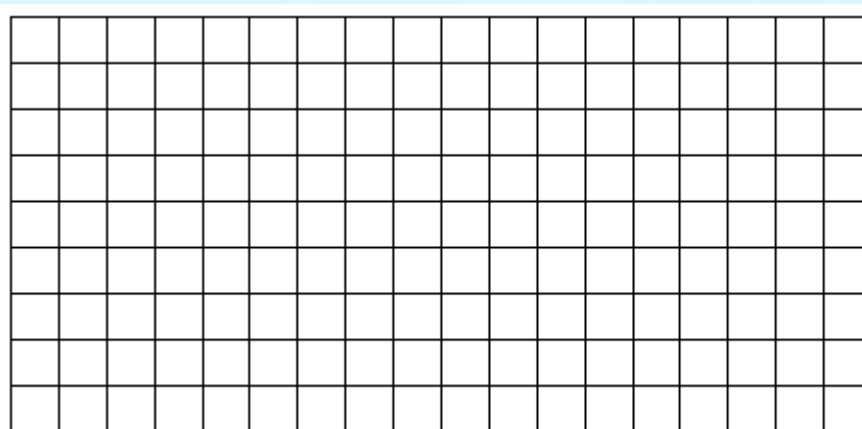
- 传真机可以通过电话线从一个地方向另一个地方传送纸质文件
- 它先将一张白底黑字的文稿扫描成1000x2000像素的图像，总共两百万像素
- 传真机将这些像素通过调制解调器发送给另一台传真机，接收端将这些像素再打印到纸张上



传真的图像一般会被压缩为原体积的1/7大小，也就是不压缩将耗费7倍时间来完成

从比特到图像

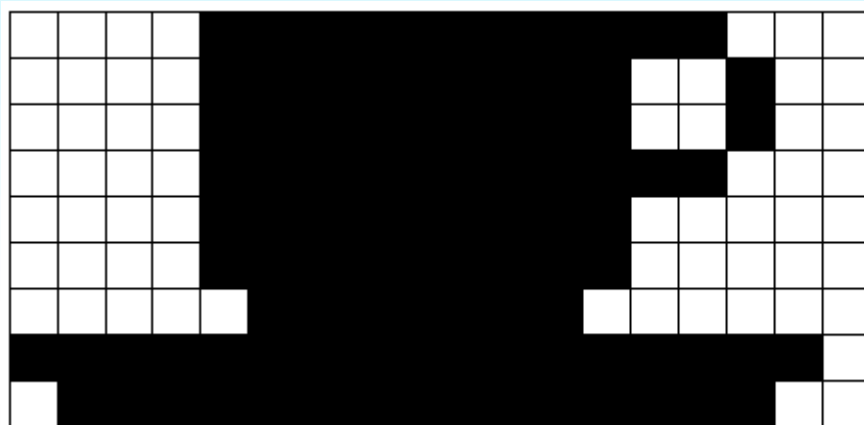
- 小游戏：图像解码



4, 11
4, 9, 2, 1
4, 9, 2, 1
4, 11
4, 9
4, 9
5, 7
0, 17
1, 15

从比特到图像

- 小游戏：图像解码



4, 11
4, 9, 2, 1
4, 9, 2, 1
4, 11
4, 9
4, 9
5, 7
0, 17
1, 15

从比特到图像

- 小游戏：图像解码

每个像素使用一个比特编码时，整张图片需要
 $18 \times 9 = 162 \text{ bits}$

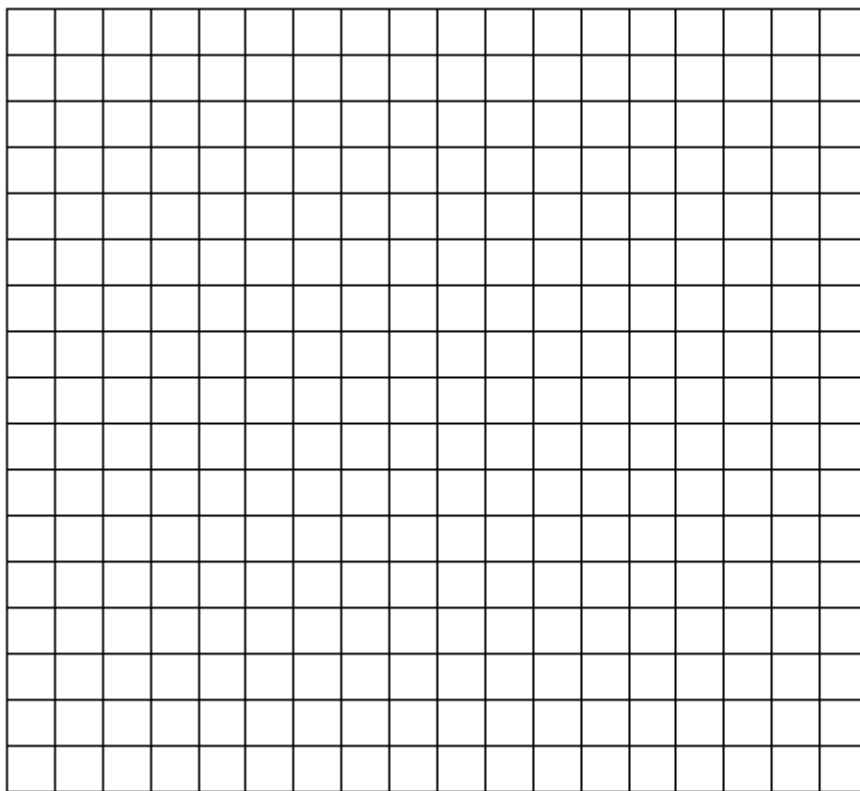


如果像素长度使用5比特来表示，那么整张图片需要
 $5 \times 22 = 110 \text{ bits}$

如果像素长度使用4比特表示，需要注意的是倒数第二行中的17需要分为两段，那么该行应该表示为0, 15, 0, 2，那么整张图片需要
 $4 \times 24 = 96 \text{ bits}$

从比特到图像

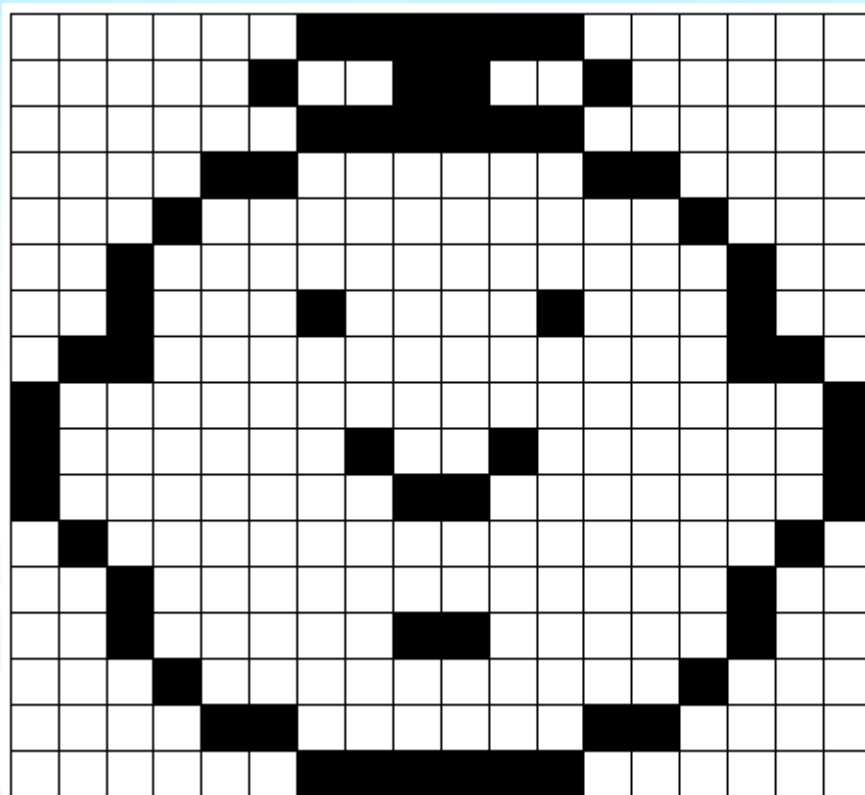
- 小游戏：图像解码



6, 2, 2, 2
5, 1, 2, 2, 2, 1
6, 6
4, 2, 6, 2
3, 1, 10, 1
2, 1, 12, 1
2, 1, 3, 1, 4, 1, 3, 1
1, 2, 12, 2
0, 1, 16, 1
0, 1, 6, 1, 2, 1, 6, 1
0, 1, 7, 2, 7, 1
1, 1, 14, 1
2, 1, 12, 1
2, 1, 5, 2, 5, 1
3, 1, 10, 1
4, 2, 6, 2
6, 6

从比特到图像

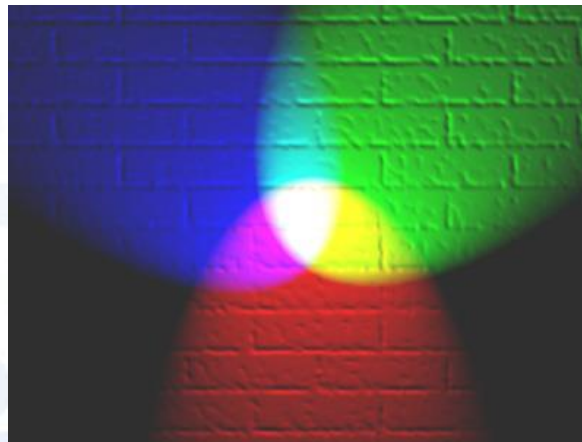
- 小游戏：图像解码



6, 2, 2, 2
5, 1, 2, 2, 2, 1
6, 6
4, 2, 6, 2
3, 1, 10, 1
2, 1, 12, 1
2, 1, 3, 1, 4, 1, 3, 1
1, 2, 12, 2
0, 1, 16, 1
0, 1, 6, 1, 2, 1, 6, 1
0, 1, 7, 2, 7, 1
1, 1, 14, 1
2, 1, 12, 1
2, 1, 5, 2, 5, 1
3, 1, 10, 1
4, 2, 6, 2
6, 6

从比特到图像

- 计算机如何表示颜色呢？
 - 计算机使用三原色光模式来表示颜色
 - 三原色模式是用三种原色——红色、绿色和蓝色的色光以不同的比例相加，以产生多种多样的色光。
 - 三原色模式也被称为RGB颜色模型



从比特到图像

- 为什么采用RGB？

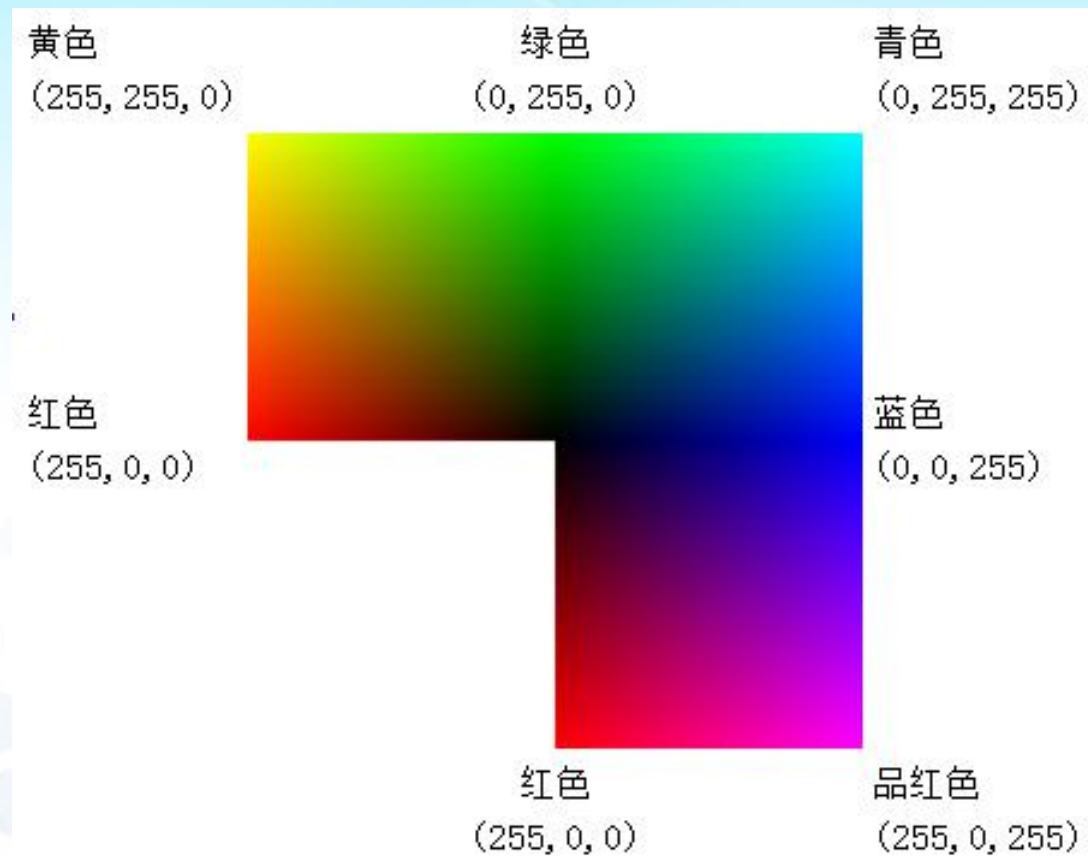
- 人眼的视网膜上有三种感色视锥细胞--感红细胞、感绿细胞、感蓝细胞
- 这三种细胞分别对红光、绿光、蓝光敏感。
- 当其中一种感色细胞受到较强的刺激，就会引起该感色细胞的兴奋，则产生该色彩的感觉。
- 人眼的三种感色细胞，具有合色的能力。
- 当一复色光刺激人眼时，人眼感色细胞可将其分解为红、绿、蓝三种单色光，然后混合成一种颜色。

从比特到图像

- 怎样编码RGB？
 - 每个像素用24比特编码 RGB 值
 - 使用三个8位无符号整数（0 到 255）表示红色、绿色和蓝色的强度
 - 这是当前主流的标准表示方法，用于真彩色和JPEG或者TIFF等图像文件格式里的通用颜色交换。
 - 它可以产生一千六百万种颜色组合，对人眼来说其中很多已经分辨不开。

从比特到图像

- 怎样编码RGB？



从比特到图像

- 图像格式
 - RAW图像格式
 - 无损压缩格式
 - BMP图像格式
 - 位图格式
 - GIF格式
 - JPEG格式
 - TIFF格式
 - PNG格式
 -