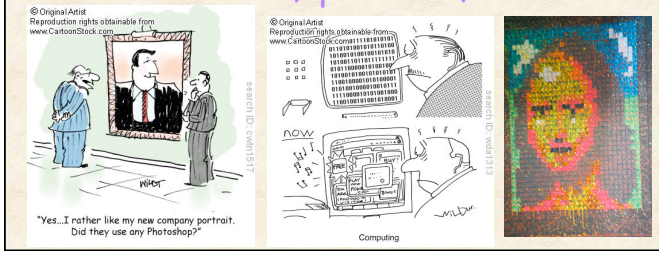
UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 202
Introduction to Computation

Professor Andrea Arpaci-Dusseau
Fall 2010

Lecture 8: How does a computer… store words, pictures, sounds?



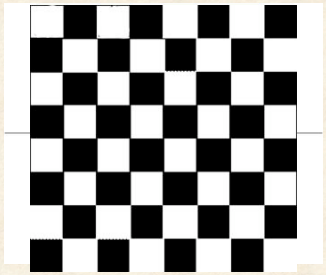
Power of Doubling

Inventor of chess so pleased king, given any prize

Cleverly asked:

- for 1st square of chess board, 1 grain
- 2nd square, 2 grains
- 3rd square, 4 grains,
- So forth…64 squares

Impossible to deliver promise!



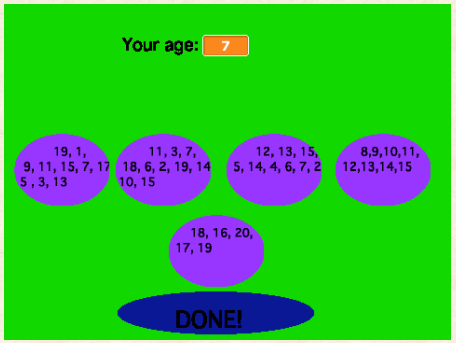
2^{63} grains on final square
2^{64}-1 grains on board
18,446,744,073,709,551,615 grains
Wheat: 80 times harvest all arable land
Rice Total volume: 37 km^3
461,168,602,000 metric tons

Why does this matter?

Adding 1 bit doubles amount can be represented

- 1 bit: 2 numbers (0, 1)
- 2 bits: 4 numbers (00, 01, 10, 11)
- 3 bits: 8 numbers (000, 001, 010, 011, 100, 101, 110, 111)
- 4 bits: 2^4 = 16 numbers
- …
- 8 bits = 2^8 = 256
- …
- 16 bits: 2^{16} = 65536
- 32 bits: 2^{32} = 4,294,967,296
- 64 bits: 2^{64} = 18,446,744,073,709,551,616

Secret Number Card Trick



Can you figure out how it works? Can you figure out how to implement it?

What kinds of data must bits represent?

Logical: True, False
- Straight-forward: Two states
- True: 1, False: 0

Numbers
- Signed, unsigned, integers, floating point, complex, rational, irrational, …

Text
- Characters, words, strings, …

Images
- Pixels, colors, shapes, movies …

Sound

Instructions

Representing Letters, Words, Strings in Binary?

How would you do it???

Representing Letters

Simplest approach: Associate each letter with binary number (table shows decimal numbers)

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

If saw in computer memory: 8 5 12 12 15 (but in binary), how would you decode (or translate from computer to human)?
hello

How would you encode (translate from human to computer): scratch
19 3 18 1 20 3 8

How many bits are needed for this simple encoding?
Round up to nearest power of two
$32 = 2^5 \rightarrow$ Need 5 bits

Problems with using only 5 bits?
Need extra characters for capitals, punctuation, spaces, etc.

ASCII: American Standard Code for Information Interchange

8 bits; how many different characters?
$2^8 = 256$

ASCII

Binary	Oct	Dec	Hex	Abbr	PR[1]	CS[2]	CEC[3]	Description
000 0000	000	0	00	NUL	␀	^@	\0	Null character
000 0001	001	1	01	SOH	␁	^A		Start of Header
000 0010	002	2	02	STX	␂	^B		Start of Text
000 0011	003	3	03	ETX	␃	^C		End of Text
000 0100	004	4	04	EOT	␄	^D		End of Transmission
000 0101	005	5	05	ENQ	␅	^E		Enquiry
000 0110	006	6	06	ACK	␆	^F		Acknowledgment
000 0111	007	7	07	BEL	␇	^G	\a	Bell
000 1000	010	8	08	BS	␈	^H	\b	Backspace[4]
000 1001	011	9	09	HT	␉	^I	\t	Horizontal Tab
000 1010	012	10	0A	LF	␊	^J	\n	Line feed
000 1011	013	11	0B	VT	␋	^K	\v	Vertical Tab
000 1100	014	12	0C	FF	␌	^L	\f	Form feed
000 1101	015	13	0D	CR	␍	^M	\r	Carriage return[4]
000 1110	016	14	0E	SO	␎	^N		Shift Out
000 1111	017	15	0F	SI	␏	^O		Shift In
001 0000	020	16	10	DLE	␐	^P		Data Link Escape
001 0001	021	17	11	DC1	␑	^Q		Device Control 1 (oft. XON)
001 0010	022	18	12	DC2	␒	^R		Device Control 2
001 0011	023	19	13	DC3	␓	^S		Device Control 3 (oft. XOFF)
001 0100	024	20	14	DC4	␔	^T		Device Control 4
001 0101	025	21	15	NAK	␕	^U		Negative Acknowledgement
001 0110	026	22	16	SYN	␖	^V		Synchronous Idle
001 0111	027	23	17	ETB	␗	^W		End of Trans. Block
001 1000	030	24	18	CAN	␘	^X		Cancel
001 1001	031	25	19	EM	␙	^Y		End of Medium
001 1010	032	26	1A	SUB	␚	^Z		Substitute
001 1011	033	27	1B	ESC	␛	^[	\e[5]	Escape[4]
001 1100	034	28	1C	FS	␜	^\		File Separator
001 1101	035	29	1D	GS	␝	^]		Group Separator
001 1110	036	30	1E	RS	␞	^^		Record Separator
001 1111	037	31	1F	US	␟	^_		Unit Separator
111 1111	177	127	7F	DEL	␡	^?		Delete[4][6]

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#
010 0100	044	36	24	$
010 0101	045	37	25	%
010 0110	046	38	26	&
010 0111	047	39	27	'
010 1000	050	40	28	(
010 1001	051	41	29	)
010 1010	052	42	2A	*
010 1011	053	43	2B	+
010 1100	054	44	2C	,
010 1101	055	45	2D	-
010 1110	056	46	2E	.
010 1111	057	47	2F	/
011 0000	060	48	30	0
011 0001	061	49	31	1
011 0010	062	50	32	2
011 0011	063	51	33	3
011 0100	064	52	34	4
011 0101	065	53	35	5
011 0110	066	54	36	6
011 0111	067	55	37	7
011 1000	070	56	38	8
011 1001	071	57	39	9
011 1010	072	58	3A	:
011 1011	073	59	3B	;
011 1100	074	60	3C	<
011 1101	075	61	3D	=
011 1110	076	62	3E	>
011 1111	077	63	3F	?

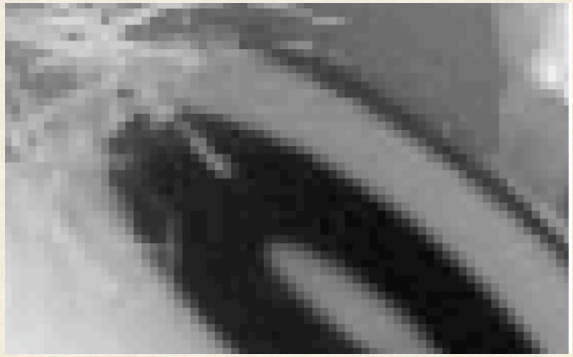
Binary	Oct	Dec	Hex	Glyph
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L
100 1101	115	77	4D	M
100 1110	116	78	4E	N
100 1111	117	79	4F	O
101 0000	120	80	50	P
101 0001	121	81	51	Q
101 0010	122	82	52	R
101 0011	123	83	53	S
101 0100	124	84	54	T
101 0101	125	85	55	U
101 0110	126	86	56	V
101 0111	127	87	57	W
101 1000	130	88	58	X
101 1001	131	89	59	Y
101 1010	132	90	5A	Z
101 1011	133	91	5B	[
101 1100	134	92	5C	\
101 1101	135	93	5D	]
101 1110	136	94	5E	^
101 1111	137	95	5F	_

Binary	Oct	Dec	Hex	Glyph
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l
110 1101	155	109	6D	m
110 1110	156	110	6E	n
110 1111	157	111	6F	o
111 0000	160	112	70	p
111 0001	161	113	71	q
111 0010	162	114	72	r
111 0011	163	115	73	s
111 0100	164	116	74	t
111 0101	165	117	75	u
111 0110	166	118	76	v
111 0111	167	119	77	w
111 1000	170	120	78	x
111 1001	171	121	79	y
111 1010	172	122	7A	z
111 1011	173	123	7B	{
111 1100	174	124	7C	|
111 1101	175	125	7D	}
111 1110	176	126	7E	~

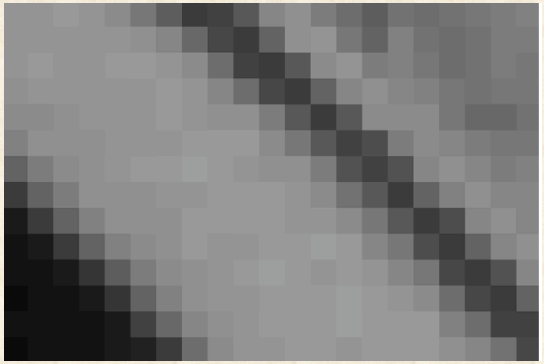
Representing Pictures

Picture looks smooth – looks analog

Picture: Zooming In



Picture: Zooming In More!

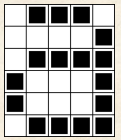


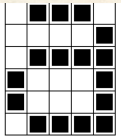
Pixel: Small square short for "Picture-Element"
We've transformed analog information into digital

How might represent picture?

Assume black and white only

How would you represent?





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

What is the problem with this approach?

Problems with Naïve Approach?

Requires many bits to represent large pictures

- Consumes a lot of storage (memory, disk)
- Consumes a lot of time to transmit (fax, network)

Important: Millions of pixels on modern displays

How can we use fewer bits?

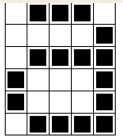
General technique: Compression

- Lossless: loses no information
- Lossy: throws away info that humans (hopefully) can't detect anyways

Run-Length Compression

Run-length encoding

- Record length of each run of white then black pixels
- Lossless or lossy?



1, 3, 1
4, 1
?, ?
0, 1, 3, 1
?, ?, ?, ?
?, ?

More Complications?

Length is represented with binary number

- Assume 8 bit binary number

Problem?

- Cannot record lengths that don't fit in 8 bits!

What is the maximum length that can be recorded?

- $2^8-1 = 255$

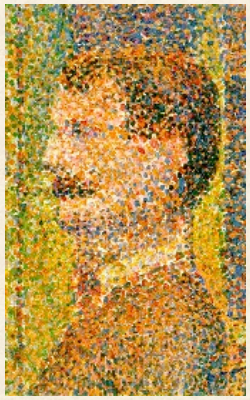
How can we represent run of 500 black pixels?

- 0 255 0 245

How would you represent colors?

- Examples: JPEG, TIFF, PNG, GIF, BMP (See Reading!)

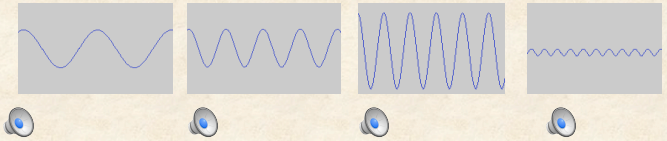
Demo: Seurat or Paintballs?



Representing Sound with Binary

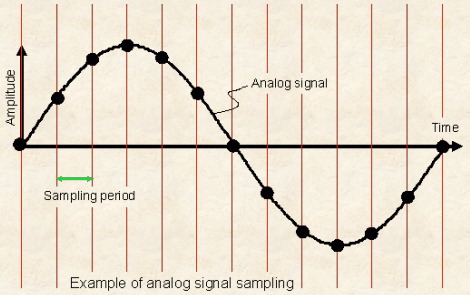
Sound is analog information: continuous sinusoidal waveform
Amplitude (height): loudness
Frequency (1/period) : pitch

- Humans can hear sounds in range 20-20,000 Hz



How would you represent? How would you transform analog into digital?

Digital representation of music



Example of analog signal sampling

Sample wave at fixed points in time

Record amplitude at those points

How would you store less information to represent this sound?
Sample less frequently (fewer points)
Use fewer bits to represent each sample

Today's Summary

Today's topics

- Text: ASCII code maps letters/symbols to numbers
- Translate analog to digital
- Images: Represent color of each pixel
  - Use run-length encoding to compress size
- Sound: Sample amplitude of sound wave

Reading:

- Pages 130-151 of "Invitation to Computer Science"

Announcements

- Homework 3: Decision tree for interative story; extend!
  - Due before class Friday
- Grades for HW1 and HW2 posted through Learn@UW