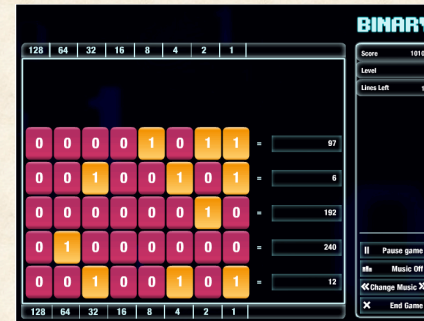


Announcements

- Homeworks 9 and 10 NOW Available
 - HW 9 Due before Thanksgiving (Tue 11/22)
 - Explore Binary Numbers and Gigapan
 - HW 10 Due after Thanksgiving (Wed 11/30)
 - Watch TED talks about TECHNOLOGY and write essay
 - Have enough info to complete both
- Exam 2 graded and returned by Monday
- No lecture Wednesday before Thanksgiving

Homework 9

Practice conversions with Tetris-like Game



http://forums.cisco.com/CertCom/game/binary_game_page.htm

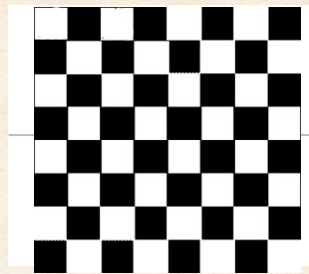
Power of Doubling

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!



2⁶³ grains on final square
 2⁶⁴-1 grains on board
 18,446,744,073,709,551,615 grains
 Wheat: 80 times harvest all arable land
 Rice Total volume: 37 km³
 461,168,602,000 metric tons

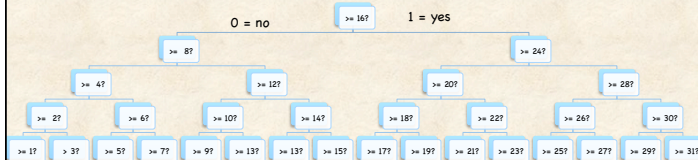
Why does this matter?

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: 2⁴ = 16 numbers
- 5 bits: 2⁵ = 32 numbers
- 8 bits = 2⁸ = 256
- ...
- 16 bits: 2¹⁶ = 65536
- 32 bits: 2³² = 4,294,967,296
- 64 bits: 2⁶⁴ = 18,446,744,073,709,551,616

Decision Tree for 0..31

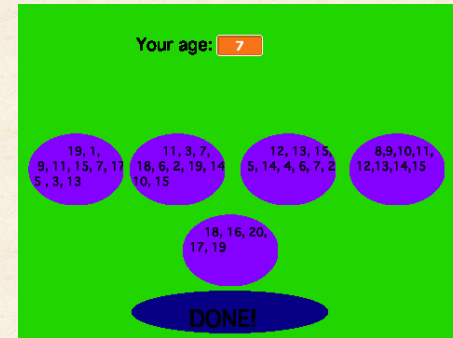
How many questions needed to find answer between 0 and 31 (32 numbers)?
 If 0 represents "no" or "false" and 1 represents "yes" or "true" what do you notice about the answers?



What is the height of this tree? (i.e. how many questions?)
 5 questions for 32 numbers; $\log_2(32) = 5$

How many bits are needed to represent number between 0..31?
 5 bits

Secret Number Card Trick



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

Other Useful Units

Bytes

- Collection of 8 bits: 1101 0011
- How many different values represented in a byte?
 - $2^8 = 256$ values
- Abbreviation: Use B for Bytes vs. b for bits

Hexadecimal numbers

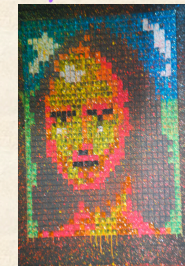
- Base-16
- Why might hexadecimal numbers be useful?
 - Two hex digits per byte
- What characters should we use?
 - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f
- 1101 0011 in hex is 0xd3

UNIVERSITY of WISCONSIN-MADISON
 Computer Sciences Department

CS 202: Introduction to Computation

Professor Andrea Arpaci-Dusseau

How does a computer...
 store words, pictures, sounds?



What kinds of data must bits represent?

Logical: True, False

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

Numbers

- Unsigned integers, signed integers, floating point

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	Cc ³	Description
000 0000	000	0	00	NUL	\	08	0	Null character
000 0001	001	1	01	SOH	\	0A	0	Start of Header
000 0010	002	2	02	STX	\	0B	0	Start of Text
000 0011	003	3	03	ETX	\	0C	0	End of Text
000 0100	004	4	04	EOF	\	0D	0	End of Transmission
000 0101	005	5	05	ENQ	\	0E	0	Enquiry
000 0110	006	6	06	ACK	\	0F	0	Acknowledgment
000 0111	007	7	07	DEL	\	0D	1	DEL
000 1000	010	8	08	BS	\	08	0	Backspace ⁴ 08 ⁵
000 1001	011	9	09	HT	\	09	0	Horizontal Tab
000 1010	012	10	0A	LF	\	0A	0	Line feed
000 1011	013	11	0B	VT	\	0B	0	Vertical Tab
000 1100	014	12	0C	FF	\	0C	0	Form feed
000 1101	015	13	0D	CR	\	0D	0	Carriage return ⁶ 0D ⁷
000 1110	016	14	0E	SO	\	0E	0	Shift Out
000 1111	017	15	0F	SI	\	0F	0	Shift In
001 0000	020	16	10	DLE	\	0F	0	Data Link Escape
001 0001	021	17	11	DC1	\	0D	0	Device Control 1 (ctrl. XON)
001 0010	022	18	12	DC2	\	0E	0	Device Control 2
001 0011	023	19	13	DC3	\	0E	0	Device Control 3 (ctrl. XOFF)
001 0100	024	20	14	DC4	\	0F	0	Device Control 4
001 0101	025	21	15	NAK	\	0D	0	Negative Acknowledgment
001 0110	026	22	16	SYN	\	0F	0	Synchronous Idle
001 0111	027	23	17	ETB	\	0F	0	End of Trans. Block
001 1000	030	24	18	CAN	\	0A	0	Cancel
001 1001	031	25	19	EM	\	0F	0	End of Medium
001 1010	032	26	1A	SUB	\	0E	0	Substitute
001 1011	033	27	1B	ESC	\	0D	0	Escape ⁸ 0D ⁹
001 1100	034	28	1C	FS	\	0C	0	File Separator
001 1101	035	29	1D	GAB	\	0E	0	Group Separator
001 1110	036	30	1E	RS	\	0A	0	Record Separator
001 1111	037	31	1F	US	\	0A	0	Unit Separator
111 1111	127	127	7F	DEL	\	7F	0	Delete ¹⁰ 7F ¹¹

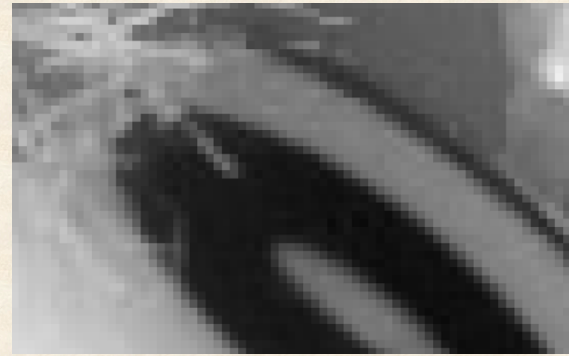
Binary	Oct	Dec	Hex	Glyph	Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20		110 0000	140	80	80	
010 0001	041	33	21	1	110 0001	141	81	81	A
010 0010	042	34	22	2	110 0010	142	82	82	B
010 0011	043	35	23	3	110 0011	143	83	83	C
010 0100	044	36	24	4	110 0100	144	84	84	D
010 0101	045	37	25	5	110 0101	145	85	85	E
010 0110	046	38	26	6	110 0110	146	86	86	F
010 0111	047	39	27	7	110 0111	147	87	87	G
010 1000	050	40	28	8	110 1000	150	88	88	H
010 1001	051	41	29	9	110 1001	151	89	89	I
010 1010	052	42	2A	A	110 1010	152	90	90	J
010 1011	053	43	2B	B	110 1011	153	91	91	K
010 1100	054	44	2C	C	110 1100	154	92	92	L
010 1101	055	45	2D	D	110 1101	155	93	93	M
010 1110	056	46	2E	E	110 1110	156	94	94	N
010 1111	057	47	2F	F	110 1111	157	95	95	O
011 0000	060	48	30		111 0000	160	96	96	P
011 0001	061	49	31	1	111 0001	161	97	97	Q
011 0010	062	50	32	2	111 0010	162	98	98	R
011 0011	063	51	33	3	111 0011	163	99	99	S
011 0100	064	52	34	4	111 0100	164	100	100	T
011 0101	065	53	35	5	111 0101	165	101	101	U
011 0110	066	54	36	6	111 0110	166	102	102	V
011 0111	067	55	37	7	111 0111	167	103	103	W
011 1000	070	56	38	8	111 1000	170	106	106	X
011 1001	071	57	39	9	111 1001	171	107	107	Y
011 1010	072	58	3A	A	111 1010	172	108	108	Z
011 1011	073	59	3B	B	111 1011	173	109	109	[
011 1100	074	60	3C	C	111 1100	174	110	110]
011 1101	075	61	3D	D	111 1101	175	111	111	^
011 1110	076	62	3E	E	111 1110	176	112	112	_
011 1111	077	63	3F	F	111 1111	177	113	113	~

Representing Pictures



Picture looks smooth - looks analog

Picture: Zooming In

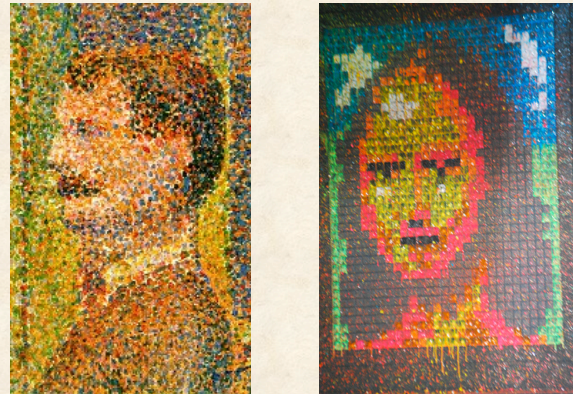


Picture: Zooming In More!



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

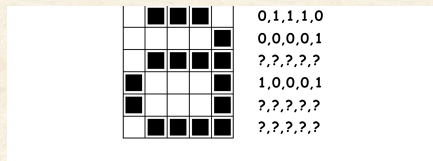
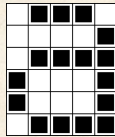
Demo: Seurat or Paintballs?



How might represent picture?

Assume black and white only

How would you represent?



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?

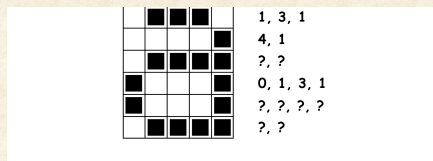
General technique: Compression

- 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?



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

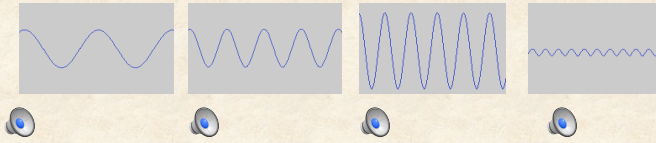
Representing Sound with Binary

Sound is **analog** information: continuous sinusoidal waveform

Amplitude (height): loudness

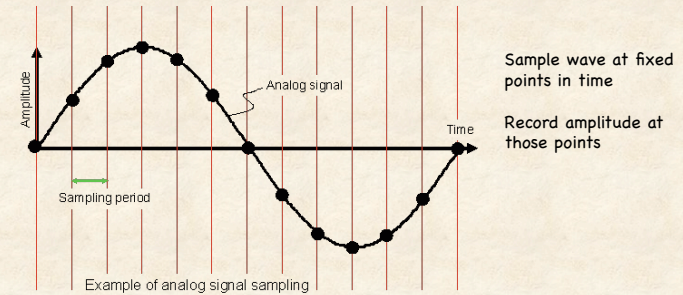
Frequency (1/period) : pitch

- 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



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

Announcements

- Next: Boolean logic (AND, OR, NOT) over Boolean Variables
- Homeworks 9 and 10 NOW Available
 - HW 9 Due before Thanksgiving (Tue 11/22)
 - Explore Binary Numbers and Gigapan
 - HW 10 Due after Thanksgiving (Wed 11/30)
 - Watch TED talks about TECHNOLOGY and write essay
 - Have enough info to complete both
- Exam 2 graded and returned by Monday
- No lecture Wednesday before Thanksgiving