Data Representation

Presented by Prapaporn Techa-Angkoon
Department of Computer Science
Faculty of Science
Chiang Mai University

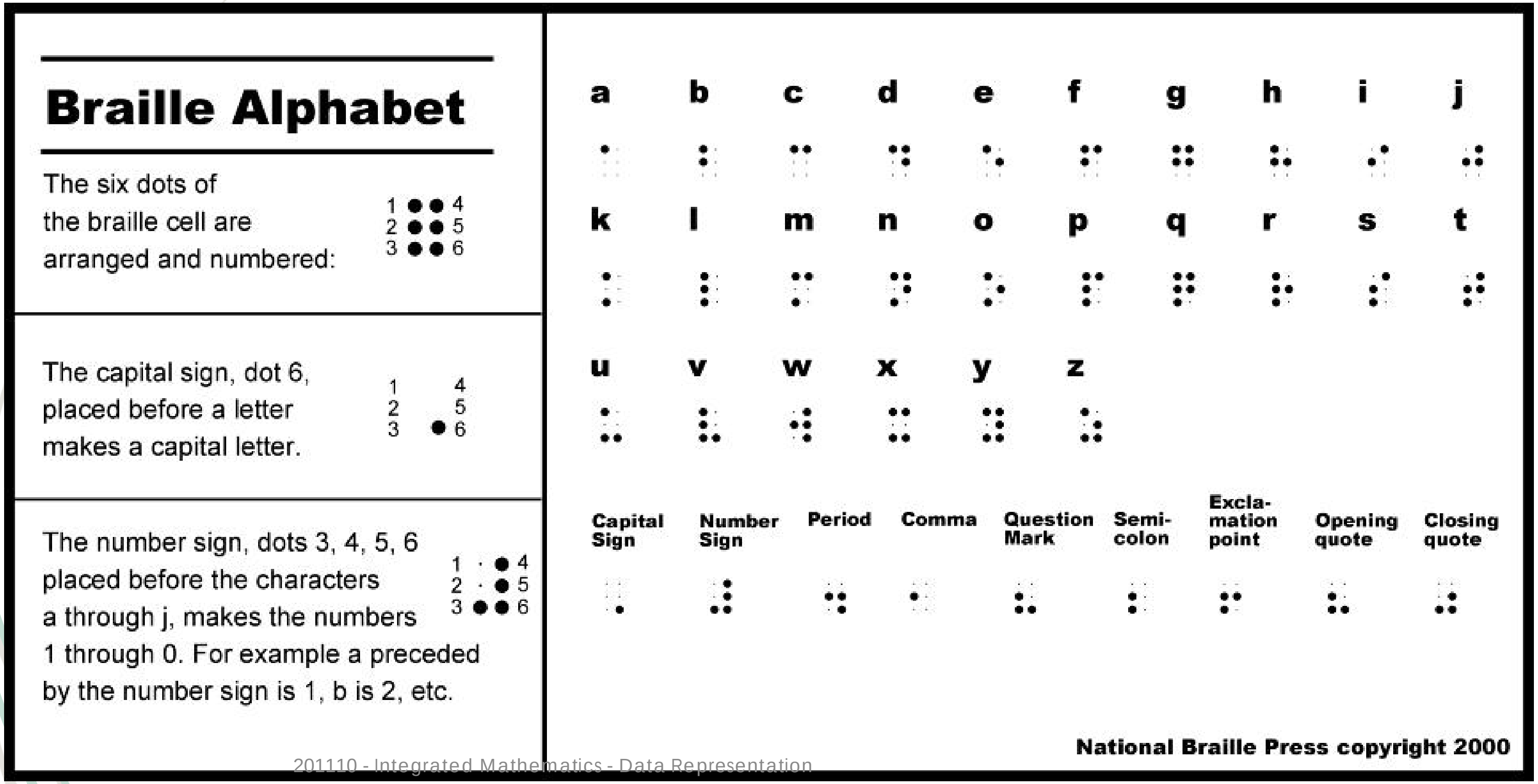
Introduction

- Data Representation refers to the methods used internally to represent information stored in a computer.
- The important types of data:
  - numbers
  - text
  - graphics of many varieties (image, video, animation)
  - sound
- ALL types of information stored in a computer are stored internally in the same simple format: a sequence of 0's and 1's.
- Computers use *binary* (base 2) number system, as they are made from binary digital components (known as transistors) operating in two states - on and off.

Introduction

- Modern computers are built up with transistors.
- Whenever an electric current pass into the transistors either an **ON** or **OFF** status will be established.
- Therefore the computer can only recognize two numbers, **0** for OFF, and **1** for ON, which can be referred to as **BIT**.
- We can use binary representations to store all kinds of data --- numbers, text, images and more.

Braille Alphabet



What is Braille?

- More than 200 years ago a 15-year-old French boy invented a system for representing text using combinations of flat and raised dots on paper so that they could be read by touch.
- The system became very popular with people who had visual impairment as it provided a relatively fast and reliable way to "read" text without seeing it.
- Louis Braille's system is an early example of a "binary" representation of data --- there are only two symbols (raised and flat).
- Each character in braille is represented with a cell of 6 dots.
- Each dot can either be raised or not raised.
- Different numbers and letters can be made by using different patterns of raised and not raised dots.

How many different patterns can be made using the 6 dots in a Braille character?

- 64 unique patterns can be made using 6 dots in Braille.
- A dot corresponds to a bit, because both dots and bits have 2 different possible values.
- The reason we're looking at Braille in this chapter is because it is a representation using bits.
- That is, it contains 2 different values (raised and not raised) and contains sequences of these to represent different patterns.
- The letter m, for example, could be written as 110010, where "1" means raised dot, and "0" means not raised dot (assuming we're reading from left to right and then down).
- This is the same as how we sometimes use 1's and 0's to show how a computer is representing data.

Telegraph Communication

Samuel F.B. Morse, 1835

A	● ▬
B	▬ ● ● ●
C	▬ ● ▬ ●
D	▬ ● ●
E	●
F	● ● ▬ ●
G	▬ ▬ ●
H	● ● ● ●
I	● ●
J	● ▬ ▬ ▬
K	▬ ● ▬
L	● ▬ ● ●
M	▬ ▬
N	▬ ●
O	▬ ▬ ▬
P	● ▬ ▬ ●
Q	▬ ▬ ● ▬
R	● ▬ ●
S	● ● ●
T	▬
U	● ● ▬
V	● ● ● ▬
W	● ▬ ▬
X	▬ ● ● ▬
Y	▬ ● ▬ ▬
Z	▬ ▬ ● ●

1	● ▬ ▬ ▬ ▬
2	● ● ▬ ▬ ▬
3	● ● ● ▬ ▬
4	● ● ● ● ▬
5	● ● ● ● ●
6	▬ ● ● ● ●
7	▬ ▬ ● ● ●
8	▬ ▬ ▬ ● ●
9	▬ ▬ ▬ ▬ ●
0	▬ ▬ ▬ ▬ ▬

Telegraph Communication

- Developed in the 1830s and 1840s by Samuel Morse (1791-1872) and other inventors, the telegraph revolutionized long-distance communication.
- It worked by transmitting electrical signals over a wire laid between stations.
- In addition to helping invent the telegraph, Samuel Morse developed a code (bearing his name) that assigned a set of dots and dashes to each letter of the English alphabet and allowed for the simple transmission of complex messages across telegraph lines.

Binary Representation

- We can use binary representations to store all kinds of data --- numbers, text, images and more.
- *Everything* stored and transmitted in our digital world is stored using just two values.
- Binary Cards
  - We use black and white cards to represent numbers.
  - The last card (on the right) to reveal that it has one dot on it.
  - The previous card, which should have two dots on it.
  - Before clicking on the next one, how many dots do you predict it will have?
  - Find a way to have exactly 22 dots

Binary Representation

http://csfieldguide.org.nz/en/interactives/binary-cards/index.html?digits=5&start=BBBBB

Binary Representation

- Binary Cards
  - You should have found that any number from 0 to 31 can be represented with 5 cards.
  - Each of the numbers could be communicated using just two words: black and white.
  - For example, 22 dots is "white, black, white, white, black"
- When we write what is stored in a computer on paper, we normally use “0” for one of the states, and “1” for the other state.
- For example, a piece of computer memory could have the following voltages:

```
low low high low high high high high low high low low
```

- We could allocate “0” to “low”, and “1” to “high” and write this sequence

```
0 0 1 0 1 1 1 1 0 1 0 0
```

Base 10 number system

- The number system that humans normally use is in base 10 (also known as decimal).
- Decimal has 10 digits -- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- A place is the place in the number that a digit is, i.e. ones, tens, hundreds, thousands, and so on.
  - For example, in the number 90328, 3 is in the "hundreds" place, 2 is in the "tens" place, and 9 is in the "ten thousands" place.
- Numbers are made with a sequence of digits.
- The right-most digit is the one that's worth the least (in the "ones" place).
- The left-most digit is the one that's worth the most.
- Because we have 10 digits, the digit at each place is worth 10 times as much as the one immediately to the right of it.

Converting to base 10

$(2561)_{10}$

Place	10^3	10^2	10^1	10^0
Digit	2	5	6	1

Base 10 $= (2\text{x}10^3) + (5\text{x}10^2) + (6\text{x}10^1) + (1\text{x}10^0)$

$= 2000 + 500 + 60 + 1 = 2561_{10}$

Representation of Numbers by Binary Code

- Computers only represent numbers using binary.
- Binary number system has two digits: 0 and 1, called *bits*.

0×2^5	+	0×2^4	+	0×2^3	+	0×2^2	+	0×2^1	+	0×2^0	=	0
0×32	+	0×16	+	0×8	+	0×4	+	0×2	+	0×1	=	0
0	+	0	+	0	+	0	+	0	+	0	=	0
0 ▾		0 ▾		0 ▾		0 ▾		0 ▾		0 ▾		

http://csfieldguide.org.nz/en/interactives/base-calculator/index.html

Converting binary number to decimal number

- **Example: 1101_2 = $(?)_{10}$**

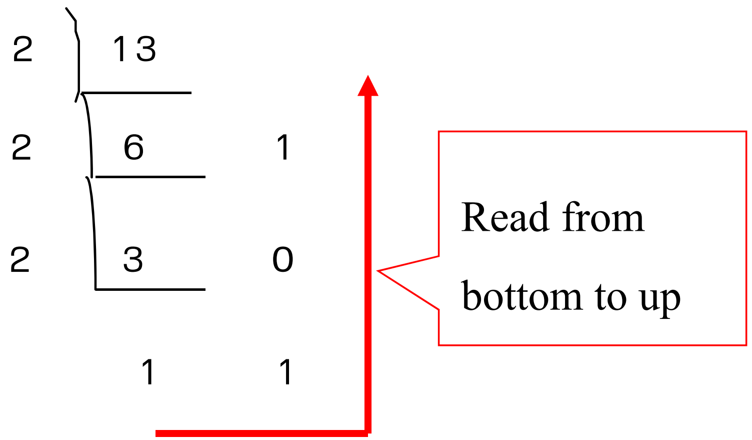
2^3	2^2	2^1	2^0	← Its Place
1	1	0	1	

= (1 x 8) + (1 x 4) + (0 x 2) + (1 x 1)
= 8+4+0+1
= 13

Converting decimal number to binary number

- ***Integer (Use MODULO – division for remainder)***

$13_{10} = (\qquad)_2$



Answer: 1101_2

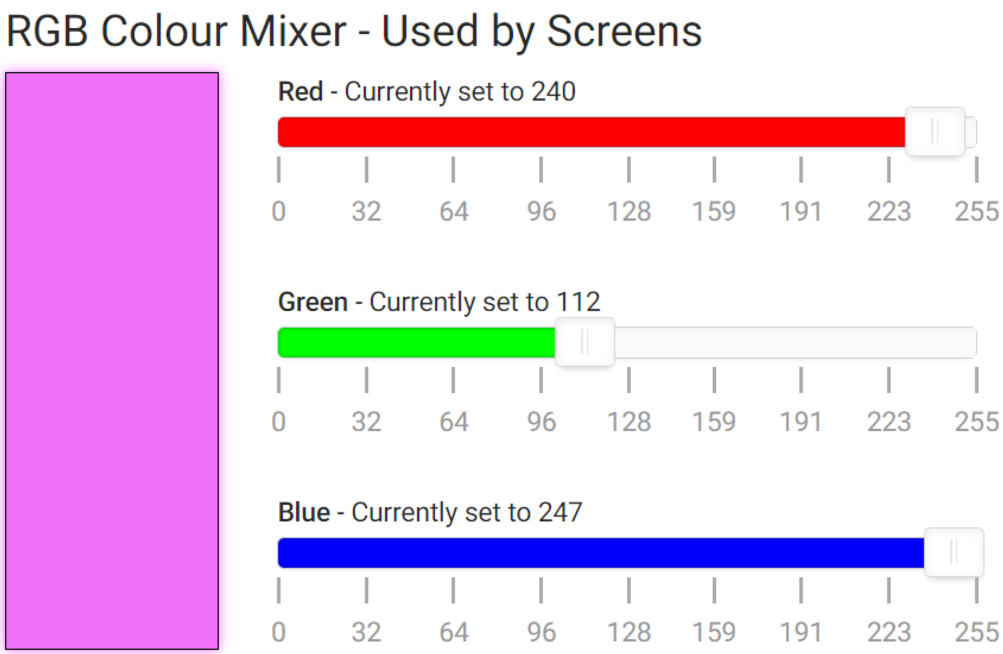
ASCII Code
(American Standard Code for Information Interchange)

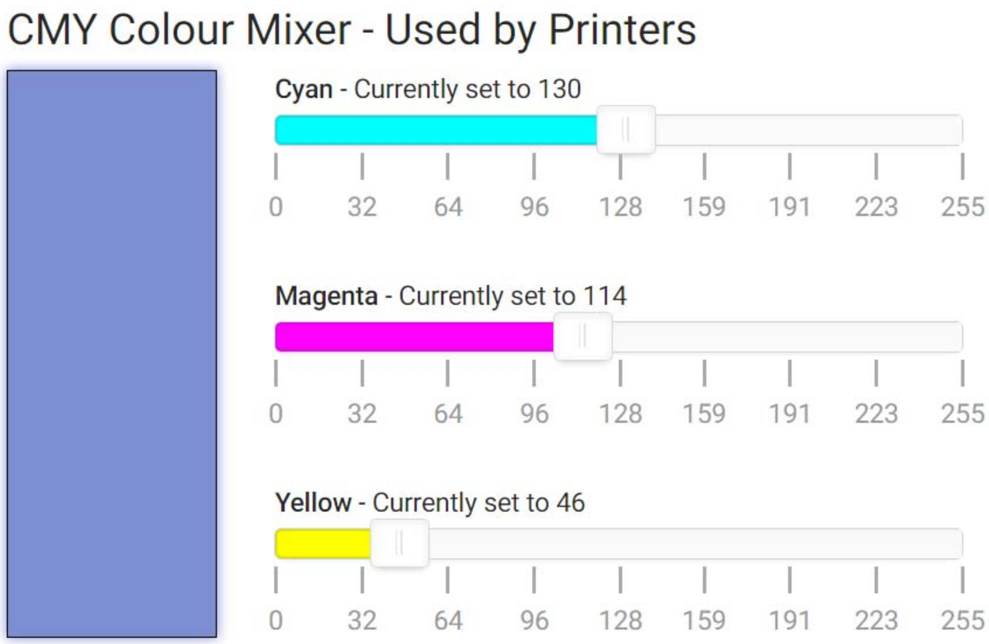
Letter	Binary	Letter	Binary
A	■□■■ ■■■□	N	■□■■ □□□■
B	■□■■ ■■□■	O	■□■■ □□□□
C	■□■■ ■■□□	P	■□■□ ■■■■
D	■□■■ ■□■■	Q	■□■□ ■■■□
E	■□■■ ■□■□	R	■□■□ ■■□■
F	■□■■ ■□□■	S	■□■□ ■■□□
G	■□■■ ■□□□	T	■□■□ ■□■■
H	■□■■ □■■■	U	■□■□ ■□■□
I	■□■■ □■■□	V	■□■□ ■□□■
J	■□■■ □■□■	W	■□■□ ■□□□
K	■□■■ □■□□	X	■□■□ □■■■
L	■□■■ □□■■	Y	■□■□ □■■□
M	■□■■ □□■□	Z	■□■□ □■□■

ASCII Code

- Computers use bits to store text.
- Text can be represented easily by assigning a unique numeric value for each symbol used in the text.
- ASCII code (American Standard Code for Information Interchange) defines 128 different symbols (all the characters found on a standard keyboard, plus a few extra), and assigns to each a unique numeric code between 0 and 127.
- In ASCII, an "A" is 65," B" is 66, "a" is 97, "b" is 98, and so forth.
- When you save a file as "plain text", it is stored using ASCII.
- ASCII format uses 1 byte per character 1 byte gives only 256 (128 standard and 128 non-standard) possible characters.
- The code value for any character can be converted to base 2, so any written message made up of ASCII characters can be converted to a string of 0's and 1's.

Color Representations





RGB : Red-Green-Blue

CMY : Cyan-Magenta-Yellow

Color Representations

- For printing, printers commonly use three slightly different primary colors: cyan, magenta, and yellow (CMY).
- All the colors on a printed document were made by mixing these primary colors.
- Both these kinds of mixing are called "subtractive mixing", because they start with a white canvas or paper, and "subtract" color from it.

Color Representations

- Computer screens and related devices also rely on mixing three colors, except they need a different set of primary colors because they are *additive*, starting with a black screen and adding color to it.
- For additive color on computers, the colors red, green and blue (RGB) are used. Each pixel on a screen is typically made up of three tiny "lights"; one red, one green, and one blue.

Describing a color with numbers

- Because a color is simply made up of amounts of the primary colors -- red, green and blue -- three numbers can be used to specify how much of each of these primary colors is needed to make the overall color.
- A commonly used scheme is to use numbers in the range 0 to 255.
- Those numbers tell the computer how fully to turn on each of the primary color "lights" in an individual pixel.
  - If red was set to 0, that means the red "light" is completely off.
  - If the red "light" was set to 255, that would mean the "light" was fully on.
- With 256 possible values for each of the three primary colors (don't forget to count 0!), that gives 256 x 256 x 256 = 16,777,216 possible colors -- more than the human eye can detect!
- Each pixel is a solid color square, and the computer needs to store the color for each pixel.

Representing a color with bits

- How many bits will we need for each color in the image?
- With 256 different possible values for the amount of each primary color, this means 8 bits would be needed to represent the number.
  - The smallest number that can be represented using 8 bits is 00000000 -- which is 0.
  - And the largest number that can be represented using 8 bits is 11111111 -- which is 255.
- Because there are three primary colors, each of which will need 8 bits to represent each of its 256 different possible values, we need **24 bits in total** to represent a color.

Representing a color with bits

- How many colors are there in total with 24 bits?
- We know that there is 256 possible values each color can take, so the easiest way of calculating it is:
  - $256 \times 256 \times 256 = 16{,}777{,}216$
  - This is the same as 2^{24}.
- Because 24 bits are required, this representation is called **24 bit color**.
- 24 bit color is sometimes referred to in settings as "True Color" (because it is more accurate than the human eye can see).
- On Apple systems, it is called "Millions of colors".

How do we use bits to represent the color?

- A logical way is to use 3 binary numbers that represent the amount of each of red, green, and blue in the pixel.
- In order to do this, convert the amount of each primary color needed to an 8 bit binary number, and then put the 3 binary numbers side by side to give 24 bits.
- The convention that the binary number for red should be put first, followed by green, and then finally blue.

How do we use bits to represent the color?

- For example, suppose you have the color that has:
  - red = 145
  - green = 50
  - blue = 123
  - You would like to represent with bits.

- Start by converting each of the three numbers into binary, using 8 bits for each.
- You should get:
  - red = 10010001
  - green = 00110010
  - blue = 01111011
- Putting these values together gives 100100010011001001111011, which is the bit representation for the color.

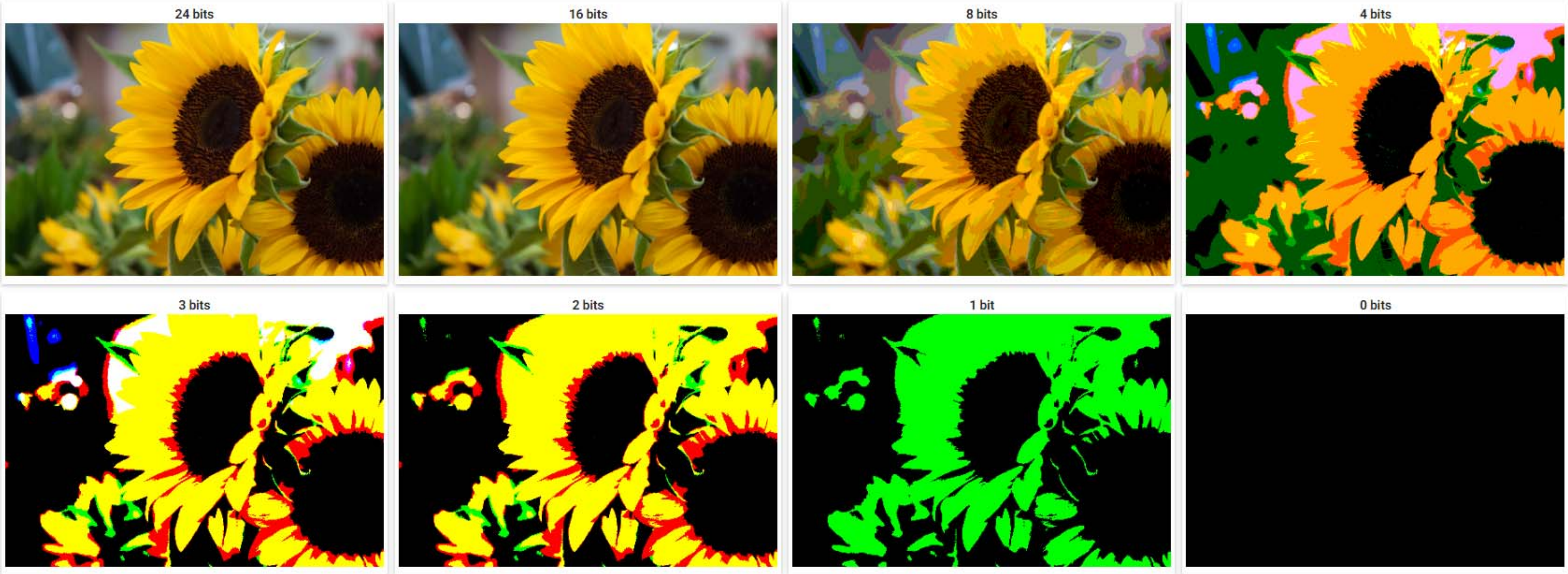
Representing colors with fewer bits

- Example:
  - 3 bits to specify the amount of red (8 possible values)
  - 3 bits to specify the amount of green (again 8 possible values)
  - 2 bits to specify the amount of blue (4 possible values)
- This gives a total of 8 bits (hence the name), which can be used to make 256 different bit patterns, and thus can represent 256 different colors.
- You may be wondering why blue is represented with fewer bits than red and green. This is because the human eye is the least sensitive to blue, and therefore it is the least important color in the representation.

Representing colors with fewer bits

- Using this scheme to represent all the pixels of an image takes one third of the number of bits required for 24-bit color, but it is not as good at showing smooth changes of colors or subtle shades, because there are only 256 possible colors for each pixel.
- The number of bits used to represent the colors of pixels in a particular image is sometimes referred to as its "color depth" or "bit depth".
  - For example, an image or display with a color depth of 8-bits has a choice of 256 colors for each pixel.

What impact does fewer bits have on the overall image?



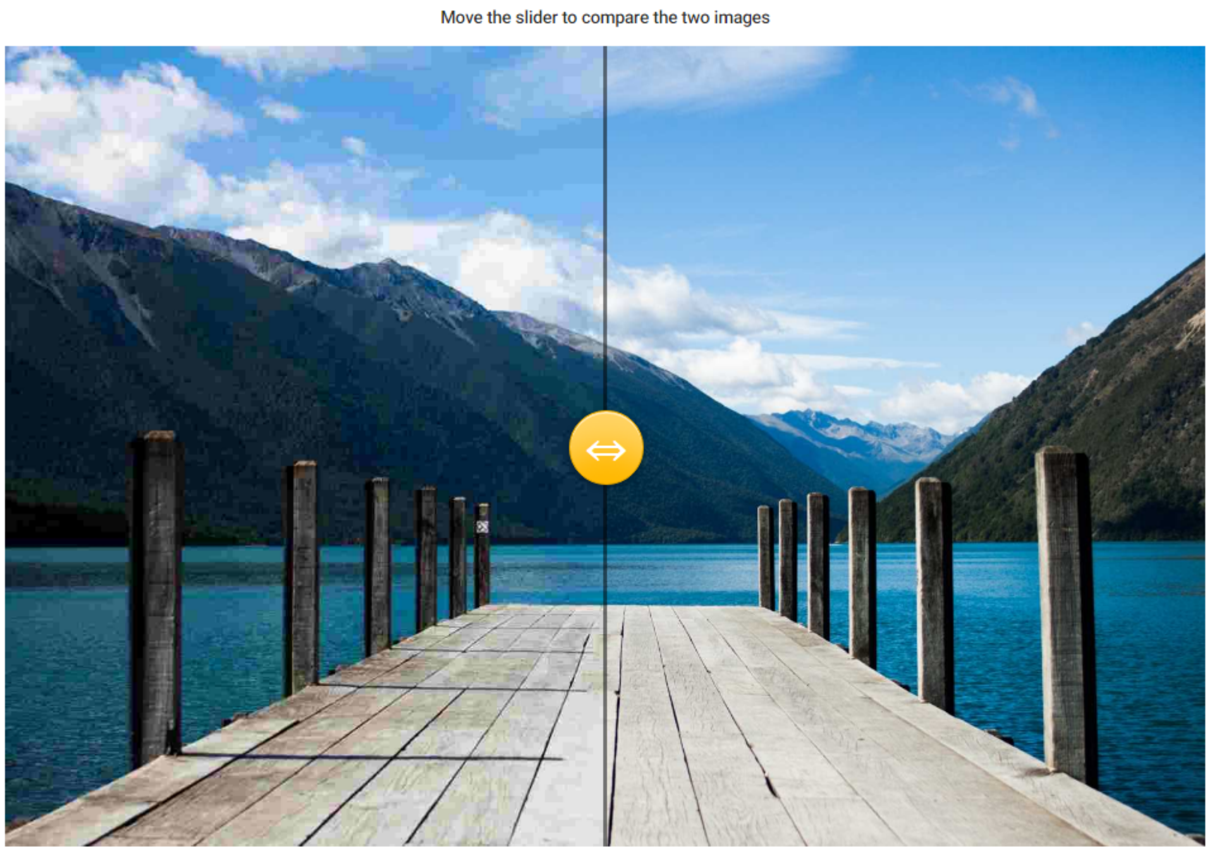
How much space will low quality images save?

- An image represented using 24-bit color would have 24 bits per pixel.
- In 600 x 800 pixel image (which is a reasonable size for a photo),
- This would contain $600 \times 800 = 480{,}000$ pixels.
- Thus, this would use $480{,}000 \times 24$ bits $= 11{,}520{,}000$ bits.
- This works out to around 1.44 megabytes.
- If we use 8-bit color instead, it will use a third of the memory, so it would save nearly a megabyte of storage.
- If the image is downloaded then a megabyte of bandwidth will be saved.

Data Compression

- If space really is an issue, then this crude method of reducing the range of colors isn't usually used; instead, compression methods such as JPEG, GIF and PNG are used.
- Data compression reduces the amount of space needed to store files.
- a highly compressed JPEG image doesn't look as sharp as an image that hasn't been compressed.
- In the data representation section, we looked at how the size of an image file can be reduced by using fewer bits to describe the color of each pixel.
- However, image compression methods such as JPEG take advantage of patterns in the image to reduce the space needed to represent it, without impacting the image unnecessarily.

Image Compression

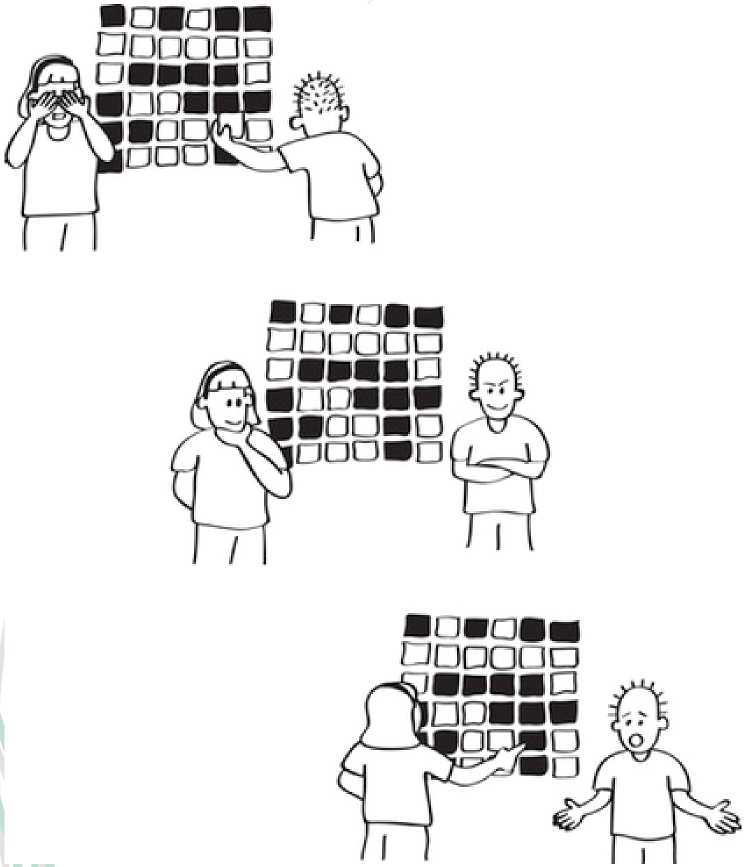


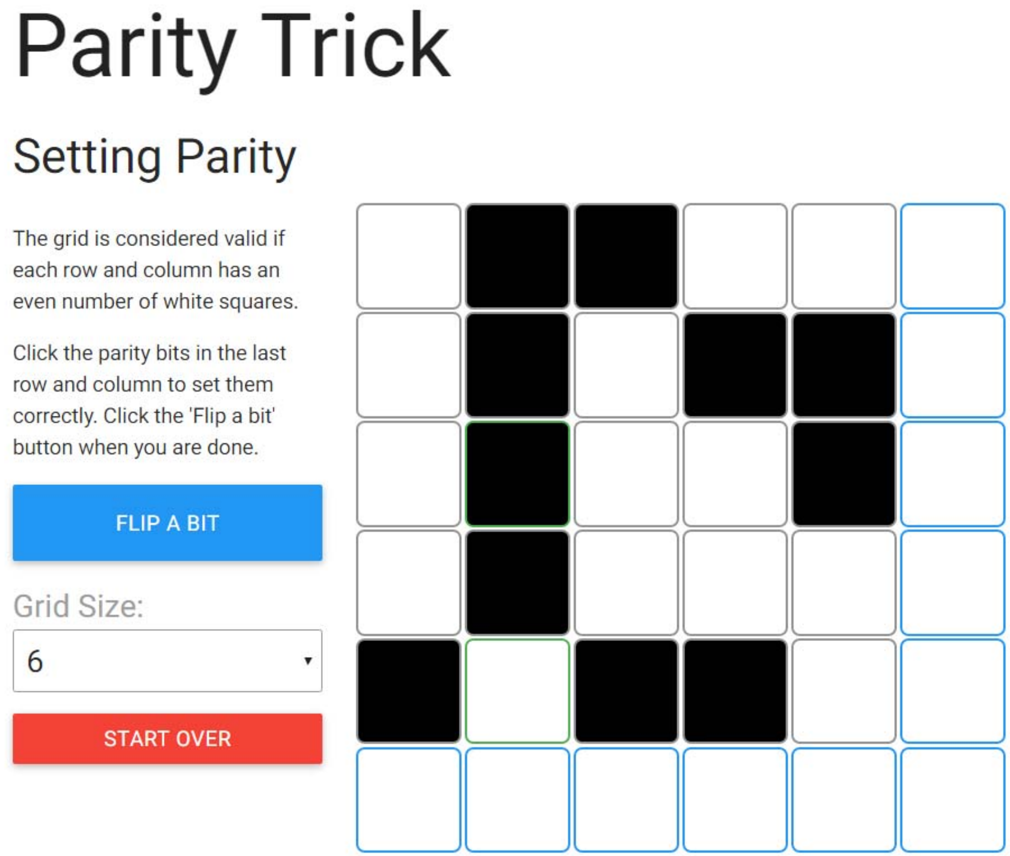
http://csfieldguide.org.nz/en/chapters/coding-compression.html

Video Compression

- Umcompressed (“raw”) video file is usually very large.
  - example: 24 frame-per-second = 24 pictures for one second of movie...
  - Need compression to reduce the file size.
- Compression in Video
  - Can use image compression to compress a single frame
  - Furthermore, frames in a time period can be very similar → compress by representing later frames with parts of earlier frames.

Data Error Detection

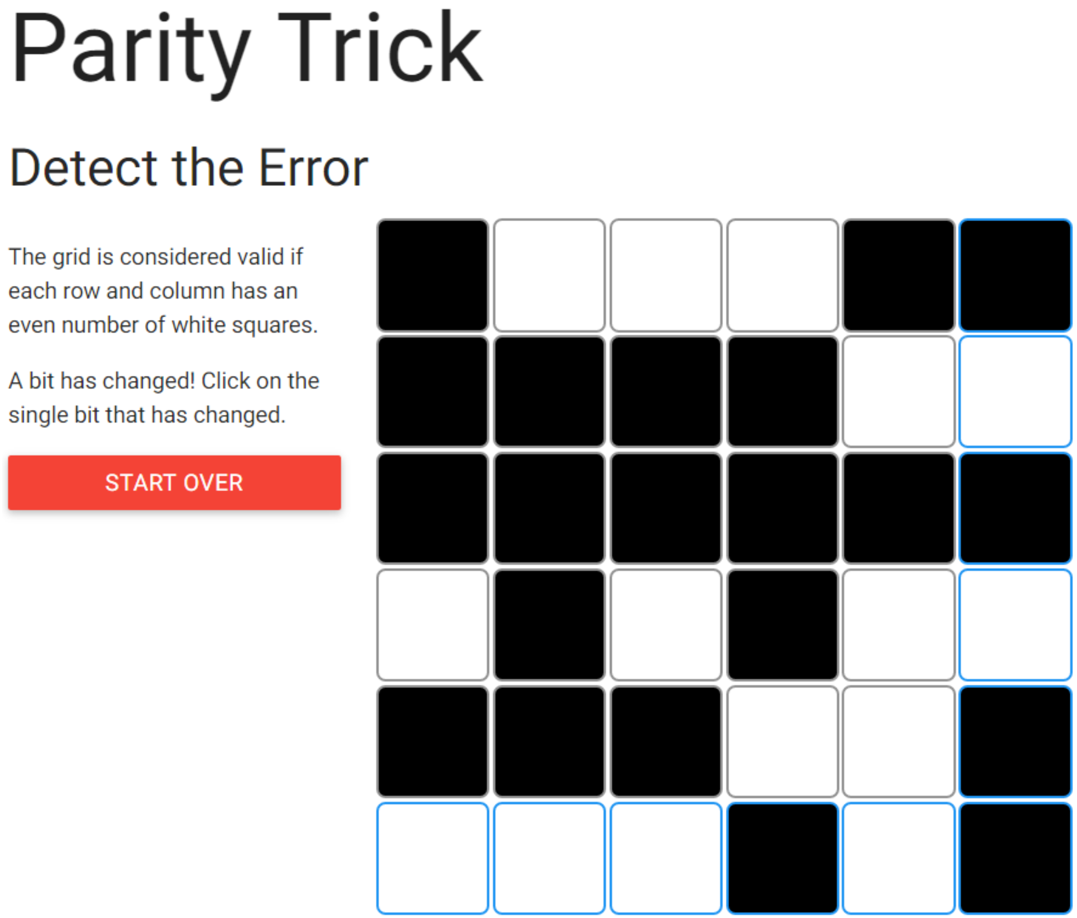




https://youtu.be/OXz64qCjZ6k
http://csfieldguide.org.nz/en/interactives/parity/index.html

Data Error Detection

- Which card was flipped?



https://youtu.be/OXz64qCjZ6k
http://csfieldguide.org.nz/en/interactives/parity/index.html

References

- http://csfieldguide.org.nz/en/chapters/data-representation.html
- http://csfieldguide.org.nz/en/chapters/coding-compression.html
- http://csfieldguide.org.nz/en/chapters/coding-error-control.html
- http://www.history.com/topics/inventions/telegraph