CHAPTER 17: INFORMATION SCIENCE

· .: ··· --- ···

Another thing we lean in this chakter is how data can be encoded so that errors can be found.

17.1 Binary Codes

How much information can you carry when using a binary code? On or off? True or false? Yes or no? Tall or short? Dot or dash?

A ***bit*** is short for ***binary digit*** and it is the basic unit of information. We will use ***0 and 1***.

A, B, C or D?

A – H?

How much information does 8 bits carry? Called a ***byte***.

EXAMPLE
A Mars lander has 16 different landing sites numbered 0 to 15. How would these be numbered in binary?

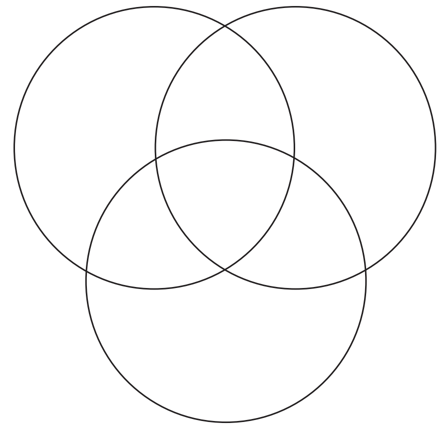
0 is ______	4 is ______	8 is ______	12 is ______
1 is ______	5 is ______	9 is ______	13 is ______
2 is ______	6 is ______	10 is ______	14 is ______
3 is ______	7 is ______	11 is ______	15 is ______

EXAMPLE

The closest Mars has been to Earth recently was 56 million km (2003). The furthest apart is about 400 million km. We want to encode check digits so our message about the landing site can correct for errors. Let c_1, c_2, and c_3 be check digits found in the following manner:

1. Place the code $a_1a_2a_3a_4$ in the circles.

2. Choose the values of c_1, c_2, and c_3 so that the sum of each circle is an even number.



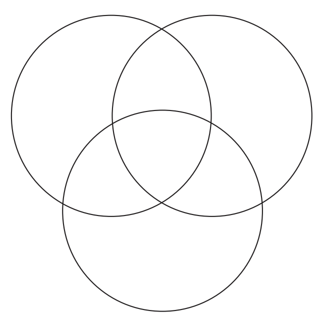
(a) What are the check digits and complete message for these sites?

0000

0110

1011

(b) Fix the error in the code 0101010 if there is only one error.



Parity refers to whether a number is odd or even. So we say even numbers have ***even parity*** and odd number have ***odd parity***.

17.2 Encoding with Parity-Check Sums

In the previous section we needed the **sum** of the numbers in each circle to have even parity by letting the check digit c be 0 or 1.

If $a_1 + a_2 + a_3$ is even, then c_1 is ____ If the sum is odd then c_1 is ____

If $a_1 + a_3 + a_4$ is even, then c_2 is ____ If the sum is odd then c_2 is ____

If $a_2 + a_3 + a_4$ is even, then c_3 is ____ If the sum is odd then c_3 is ____

The sums $a_i + a_j + a_k$ are called ***parity-check sums***.

A set of words composed of 0's and 1's that has a message and parity check sums appended to the message is called a ***binary linear code***. The resulting strings are called ***code words***.

The process of determining the message you were sent is called ***decoding***. If you are sent a message v and receive the message as u, how can it be decoded?

The ***distance between two strings*** of equal length is the number of positions in which the strings differ.

<u>*EXAMPLE*</u>
Find the distance between the given pairs of strings.

(a) 1101 and 1101

(b) 10001 and 11001

(c) 01010101 and 10101010

When decoding a message, decode *u* as the message that differs from *v* in the fewest number of positions. If there is a tie, don't decode.

The ***nearest neighbor decoding method*** decodes a message as the code word that agrees with the message in the most positions provided there is only one such message.

EXAMPLE

The table below provides the code words for all 16 landing sites. Use this table to decode the given messages.

0	0000000	5	0101110	11	1011010
1	0001011	6	0110010	12	1100011
2	0010111	7	0111001	13	1101000
3	0011100	8	1000110	14	1110100
4	0100101	9	1001101	15	1111111
		10	1010001		

(a) 0001000

(b) 0010010

0000000 0010010	0100101 0010010	1000110 0010010	1100011 0010010
0001011 0010010	0101110 0010010	1001101 0010010	1101000 0010010
0010111 0010010	0110010 0010010	1010001 0010010	1110100 0010010
0011100 0010010	0111001 0010010	1011010 0010010	1111111 0010010

EXAMPLE

If we wanted to encode the English alphabet, we would need at least 26 different codes. How many bits are needed?

Longer codes need more check digits. And do we wish to just determine if there are errors or do we wish to also be able to correct errors?

The ***weight of a binary code*** is the minimum number of 1's that occur among all non-zero code words of that code.

EXAMPLE

What is the weight of the code below?

C = {0000000, 0001011, 0010111, 0011100, 0100101, 0101110, 0110010, 0111001, 1000110, 1001101, 1010001, 1011010, 1100011, 1101000, 1110100, 1111111}

Consider a code of weight t,

- The code can detect $t-1$ or fewer errors.
- If t is odd, the code will correct $(t-1)/2$ or fewer errors.
- If t is even, the code will correct any $(t-2)/2$ or fewer errors.

Data Compression

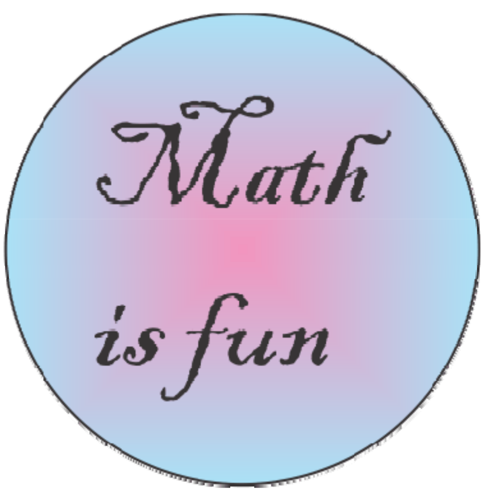
Another binary code is the Morse code. The table below shows how often various letters occur in a typical text written in English:

	A	B	C	D	E	F	G	H	I	J	K	L	M
Percentage:	8	1.5	3	4	13	2	1.5	6	6.5	0.5	0.5	3.5	3
	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Percentage:	7	8	2	0.25	6.5	6	9	3	1	1.5	0.5	2	0.25

The next table shows the Morse code. What do you notice?

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	——··

Data compression is the process of encoding data so that the most frequently occurring data are represented by the fewest symbols.



ImageExample.gif is 34 KB
compuserve bitmap, 256 colors

ImageExample.png is 42 KB
portable network graphics, RGB 24-bit color

ImageExample.bmp is 142 KB
windows bitmap, paletted 8-bit color

ImageExample.jpg is 588 KB
JPEG bitmat, CMYK 32-bit color. High (80%) quality

A ***compression algorithm*** converts data from an easy-to-use format to one that is more compact. MP3, for example.

EXAMPLE

DNA is made from four bases: adenine (A), cytosine (C), guanine (G) and thymine (T).

(a) How could these 4 bases be represented in binary?

(b) Encode the sequence AACGCAT

(c) Given that A occurs the most often followed by C, T and G, use the encoding

$A \rightarrow 0 \quad C \rightarrow 10 \quad T \rightarrow 110 \quad G \rightarrow 111$

Encode the sequence AACGCAT

(d) Decode the sequence 1001100111100

Delta function encoding uses the differences in one value to the next to encode the data.

EXAMPLE

Use delta function encoding to compress the daily high temperatures in College Station for the first 10 days of August 2011:

106	105	106	105	105	103	102	104	105	104

Huffman coding is a way to assign shorter code words to those characters that occur more often. Consider the case of the following 6 characters that occur with the given probabilities:

G	H	I	J	K	L
0.06	0.13	0.23	0.17	0.20	0.21

1. Arrange these letters from least to most likely:

2. Add the probabilities of the two least likely characters and combine them. Keep the letter with the smaller probability on the left. Arrange the new list from least to most likely.

3. Repeat until there is only one arrangement of characters.

4. To assign a binary code to each letter, display the information in a tree assigning 0 to the branch with the lower probability.

Decode 1011011110011001111 0

17.3 Cryptography

The process of disguising data is called encryption. Cryptology is the study of making and breaking secret codes.

A ***Caesar cipher*** shifts the letters of the alphabet by fixed amount.

EXAMPLE
Create a Caesar cipher that shifts the alphabet by 6 letters and use it to encrypt the message MATH IS FUN.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
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

Note that the Caesar cipher used replaces letter n with $(n+x)$ mod 26.

A ***decimation cipher*** multiplies the position of each letter by a fixed number k (called the ***key***) and then uses modular arithmetic. To use a decimation cipher,

1. Assign the letters A – Z to the numbers 0 – 25.
2. Choose a value for the key, k, that is an odd integer from 3 to 25 but not 13 (why not?)
3. Multiply the value of each letter (i) by the key (k) and find the remainder when divided by 26. That is $x = ki \bmod 26$.
4. To decrypt a message, the encrypted value x needs to be multiplied by the decryption letter j such that $kj = 1 \bmod 26$ and then the remainder mod 26 is the original letter.

EXAMPLE

Encrypt the message MATH IS FUN using the key 7.

M A T H I S F U N

What would be the decryption key?

EXAMPLE

The message below was encrypted with the key 21. The decryption key is j = 5. What does the message say?

S A T I I N I E J

A ***Vigenère cipher*** uses a ***key word*** to encode the characters.

EXAMPLE
Encrypt the message MATH IS FUN using the key word *BOX*.

M A T H I S F U N

EXAMPLE
Decode the message EMYWGIRII that used the key word *RICE*.

E M Y W G I R I I

To carry out binary addition, add each of the digits. If the result is even, enter 0. If the result is odd, enter 1.

EXAMPLE
Add the codes 100011 and 110010.

EXAMPLE
What happens when a binary code is added to itself?

EXAMPLE
What are some encryption methods used on the internet?