9 THEORY OF CODES

After studying this chapter you should

- understand what is meant by noise, error detection and correction;
- be able to find and use the Hamming distance for a code;
- appreciate the efficiency of codes;
- understand what is meant by a linear code and parity-check matrix;
- be able to decode a transmitted word using the parity check matrix.

9.0 Introduction

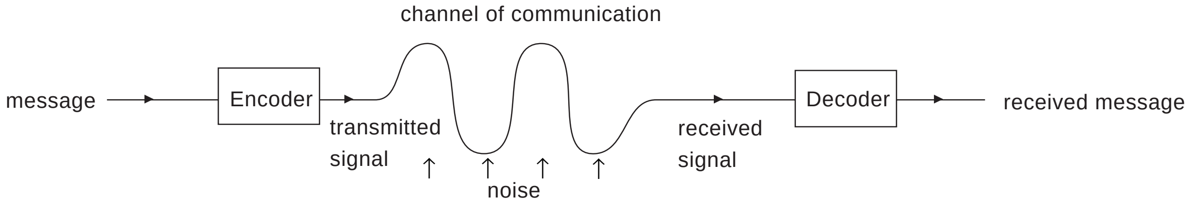
In this chapter you will look in some depth at the way in which codes of varying construction can be used both to detect and sometimes correct errors. You have already seen in Chapter 8 how check digits for ISBN numbers and bar codes are used to detect errors. In this chapter you will look at codes relevant to data transmission, for example the transmission of pictures from Mars to the Earth, and see how such codes are designed.

9.1 Noise

To take an example, in TV broadcasting the message for transmission is a picture in the studio. The camera converts this into a 625-row array of packages of information, each package denoting a particular colour. This array, in the form of an electrical signal, is broadcast via antennae and the atmosphere, and is finally interpreted by the receiving set in the living room. The picture seen there differs somewhat from the original, errors having corrupted the information at various stages in the channel of communication. These errors may result in effects varying from subtle changes of colour tone to what looks like a violent snowstorm. Technically, the errors are all classified as **noise**.

What form does 'noise' take in telephone calls?

A model of data transmission is shown below.



Normally, the message is encoded, the signal transmitted to the receiver, and then decoded with a received message. It is in the transmission that noise can affect the signal.

For example, the Mariner 9 spacecraft in 1971 sent television pictures of the planet Mars across a distance of 84 million miles. Despite a very low power transmitter, the space-probe managed to send data which eventually resulted in very high quality pictures being shown on our screens. This was in part largely due to the sophisticated coding system used.

As a very simple example, consider a code which has four **codewords**:

$$C = \{(0\,0),\ (0\,1),\ (1\,0),\ (1\,1)\}$$

Each codeword has **length** 2, and all digits are either 0 or 1. Such codes are called **Binary Codes**.

Could you detect an error in the transmission of any of these codewords?

One way to detect an error, would be to repeat each codeword, giving a new code

$$C_1 = \{(0\,0\,0\,0),\ (0\,1\,0\,1),\ (1\,0\,1\,0),\ (1\,1\,1\,1)\}$$

Here each pair of digits is repeated.

Can Code C_1 detect a single error?

For example, if the codeword (0 1 0 1) was corrupted to (1 1 0 1) it is clear that an error can be detected, as (1 1 0 1) is **not** one of the codewords.

Can a single error in a codeword be corrected?

This is not as straightforward to answer since, for example, (1 1 0 1) could have also been (1 1 1 1) with one error, as well as (0 1 0 1). So this code can detect a single error but cannot correct it. It should also be added that the **efficiency** (or rate) of this code is given by

$$\frac{\text{number of original message bits}}{\text{length of codeword}} = \frac{2}{4} = \frac{1}{2},$$

since each codeword in the original message had only two digits (called **bits**).

Activity 1

Consider a code designed to specify one of four possible directions

up	down	left	right
(0 0 0)	(1 1 0)	(0 1 1)	(1 0 1)

Can this code detect any single error made during the transmission of a codeword? Can it correct it?

Often codes include a **parity check** so that, for example, the code C is transformed to C_2 as shown below.

C	C_2
0 0	0 0 0
0 1	0 1 1
1 0	1 0 1
1 1	1 1 0

The extra last digit in C_2 (or check digit) is 0 if the sum of the digits modulo 2 is zero (or the number of 1's is even), or is 1 if the sum of the digits modulo 2 is 1 (or the number of 1's is odd). (Modulo 2 means $0+0=0$, $0+1=1$, $1+0=1$, $1+1=0$.)

Can Code C_2 detect errors now?

Using the previous definition, the efficiency of Code C_2 is $\frac{2}{3}$.

None of the codes considered so far can correct errors.

Activity 2

Design a code containing 4 codewords, each of length 5, which can detect and correct a single error.

9.2 Error correction

Clearly codes which can both detect and correct errors are of far greater use – but the efficiency will decrease, since extra essentially redundant information will have to be transmitted.

For example, here is a code that can be used to idenfy four directions:

up	down	left	right
(0 0 0 0 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(1 1 0 0 1 1)

The length of each codeword is 6, but since the number of message bits is essentially 2, i.e. the code could consist of

(0 0), (1 1), (0 1), (1 0)

and its efficiency is $\frac{2}{6} = \frac{1}{3}$. But, as you see, it can **correct** single errors.

All the codes in this Chapter are binary codes consisting of just 2 code symbols, 1 and 0. The number of codewords in a full code is always a power of 2, for example, 2^k, where k is called the dimension of the code. k is the essential number of message bits in the code. There are 4 codewords in the code and since $4 = 2^2$, the dimension is 2 and the essential number of message bits is 2.

Activity 3

The following words from the above code have been received. Assuming that only one error has been made in the transmission of each codeword, determine if possible the actual codeword transmitted:

(a) (1 0 0 0 0 0) (b) (1 1 0 0 0 0) (c) (0 1 0 0 1 1)

Can the code above detect if 2 errors have been made in the transmission of a codeword?

Activity 4 Codes

Consider Code 5 given in Appendix 5. Find out how many errors this code can detect and correct by considering, for example, words such as

(a) (1 1 0 0 0 0 0) (b) (0 1 1 1 1 1 1) (c) (1 0 0 0 1 0 0)

which are in error.

By now you should be beginning to get a feel for what is the important characteristic of a code for the determination of the errors that can be detected and corrected. The crucial concept is that of **distance** between codewords.

The **distance** between any two codewords in a code is defined as the number of actual differences between the codewords; for example

$$d\left((111),\ (010)\right) = 2,$$

since the first and third digit are different;

whilst $\quad d\left((0101),\ (1011)\right) = 3.$

The **Hamming distance** is defined as the **minimum** distance between any two codewords in the code and is usually denoted by δ.

Example

Determine the Hamming distance for the code with codewords

(1 1 0 0 0), (0 0 1 0 1), (1 0 1 0 1), (1 1 1 1 1)

Solution

You must first find distances between all the codewords.

$$d\left((11000),\ (00101)\right) = 4$$

$$d\left((11000),\ (10101)\right) = 3$$

$$d\left((11000),\ (11111)\right) = 3$$

$$d\left((00101),\ (10101)\right) = 1 \quad \leftarrow \text{Hamming distance } \delta = 1 \text{ (minimum of 1, 2, 3 and 4)}$$

$$d\left((00101),\ (11111)\right) = 3$$

$$d\left((10101),\ (11111)\right) = 2$$

Why is the Hamming distance crucial for error detection and correction?

Activity 5 Hamming distance

Determine the Hamming distance for

Codes 1, 2, 3, 4 and **5**

given in Appendix 5.

To try and see the connection between the Hamming distance, δ, and the number of errors that can be detected or corrected, you will consider Codes 1 to 5 from Appendix 5.

Activity 6

Copy and complete this table.

Code	Hamming distance	Errors corrected	Errors detected
1	2	0	1
2	3	1	1
3	...	...	...
4	...	...	...
5	...	...	...

Also add on to the table any other codes considered so far. Can you see a pattern?

The first thing that you probably noticed about the data in the table for Activity 6 is that the results are different depending on whether *n*, the number of bits, is even or odd. It looks as if, for

$\delta = 2$, you can detect 1 error but correct 0 errors

$\delta = 3$, you can detect 1 error and correct 1 error.

How do you think the pattern continues for $\delta = 4$ and 5?

Activity 7

Construct a code for which $\delta = 4$. How many errors can it detect or correct? Similarly, construct a code for which $\delta = 5$ and again determine how many errors it can detect or correct. Can you suggest a generalisation of the results?

As can be seen from Activities 6 and 7, there is a distinct pattern emerging. For

δ odd,	the code can correct and detect up to $\frac{1}{2}(\delta - 1)$ errors.
δ even,	the code can correct up to $\frac{1}{2}(\delta - 2)$ errors, and detect up to $\frac{1}{2}\delta$ errors.

Activity 8 Validating the results

Check that the above result holds for Code 9 in Appendix 5.

Exercise 9A

1. The '2 out of 5' code consists of all possible words of length 5 which have exactly two 1 s; for example, (1 0 1 0 0) belongs to the code but (1 1 0 1 0) does not.

   List all possible codewords and explain why this code is particularly useful for the transmission of numeric data. What is the Hamming distance for this code?

2. Analyse the '3 out of 7' code, defined in a similar way to the '2 out of 5' code in Question 1. Determine its Hamming distance and hence find out how many errors it can detect and correct.

3. Determine the Hamming distance for Code 7 in Appendix 5. Hence find out how many errors this code can detect and correct.

4. Show that the code

   $C_4 = \{(00000), (11000), (00011), (11111)\}$

   can detect but not correct single errors in transmission.

9.3 Parity check matrix

The main challenge of coding theory is to find good effective codes - that is, ones which transmit information efficiently yet are able to detect and correct a suitable number of errors.

Remember that the length of a code is the number of bits of its codewords - this is usually referred to as length n.

The codes used in the previous unit and those constructed here add another $(n-k)$ check bits to each message of length k to make a codeword of length n. For example, Code 3

0 0 0 0

0 1 0 1

1 0 1 0

1 1 1 1

is found by adding two check digits to each codeword of length 2, namely

0 0

0 1

1 0

1 1

So here $k=2$, $n=4$, and there are $n-k = 4-2 = 2$ check digits. The number k is called the **dimension** of the code and, as you saw in Section 9.1, the **efficiency** (or rate) is given by $\frac{k}{n}$.

A code of length n, with k message bits, is called an (n, k) code.

Example

Show that Code 4 is a (4, 2) code.

Solution

Since the codewords of Code 4 are

0 0 0 0

1 1 0 0

0 0 1 1

1 1 1 1

then $n=4$, and $k=2$ since two columns are repeated. (Alternatively you might like to think of it in terms of 4 codewords, which could be coded using codewords of length 2; e.g. (0 0), (1 1), (1 0), (0 1); thus $k=2$ and two more bits have been added to give $n=4$.)

Thus Code 4 is a (4, 2) code.

A vector and matrix notation will be adopted, writing a codeword **x** as a row vector; for example [1 1 0 1] .

The transpose of **x** is a column vector, $\mathtt{x}' = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$

Let **x** be a codeword with *n* bits, so that

$$\mathtt{x} = [x_1\ x_2\ \dots\ x_n] \qquad (1 \times n \text{ matrix})$$

and

$$\mathtt{x}' = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

For an (n, k) code, a **parity check matrix**, H, is defined as an $(n-k)\times n$ matrix such that

$$\mathrm{H}\,\mathtt{x}' = \mathtt{0} \ \ (\text{modulo } 2)$$

and when no row of H consists just of zeros.

An alternative way of finding *k* is to write (when possible) the number of codewords in the code as a power of 2. This power is *k*, the dimension;

i.e. no. of codewords = 2^k.

Check this result for Code 3 and Code 4.

There are 4 codewords, so $2^k = 4 = 2^2$ and hence $k=2$.

Example

Show that $\mathrm{H} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

is a parity check matrix for Code 4.

Solution

For Code 4, $\mathrm{x}_1 = [0\ 0\ 0\ 0]$, $\mathrm{x}_2 = [1\ 1\ 0\ 0]$, $\mathrm{x}_3 = [0\ 0\ 1\ 1]$, $\mathrm{x}_4 = [1\ 1\ 1\ 1]$

and

$$\mathrm{H}\,\mathrm{x}_1' = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathrm{H}\,\mathrm{x}_2' = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1+1 \\ 1+1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

(arithmetic is modulo 2)

$$\mathrm{H}\,\mathrm{x}_3' = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1+1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathrm{H}\,\mathrm{x}_4' = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1+1 \\ 1+1+1+1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Hence H is a parity check matrix for Code 4.

In fact, the codewords $\mathrm{x} = (x_1\ x_2\ x_3\ x_4)$ of Code 4 are precisely the solutions of

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow \begin{cases} x_1 + x_2 = 0 & \text{(modulo 2)} \\ x_1 + x_2 + x_3 + x_4 = 0 & \text{(modulo 2)} \end{cases}$$

Activity 9

Find all solutions, modulo 2, of the two equations above.

This property, that $\mathrm{H}\,x' = 0$ has as its solution the codewords of the code, leads us into a method of finding a parity check matrix. Consider for example Code 2 from Appendix 5. This has length 6 and all codewords in Code 2 satisfy

$$x_2 + x_3 + x_4 = 0 \quad \text{(modulo 2)}$$
$$x_1 + x_3 + x_5 = 0 \quad \text{(modulo 2)}$$
$$x_1 + x_2 + x_6 = 0 \quad \text{(modulo 2).}$$

In matrix form these can be written as

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Since $n = 6$, and $k = 3$, the above 3×6 matrix H is a parity check matrix for Code 2.

Example

Find a parity check matrix for Code 3 from Appendix 5.

Solution

For Code 3, $n = 4$ and $k = 2$, so it is a $(4, 2)$ Code and H will be a 2×4 matrix. Now for all codewords in Code 3,

$$x_1 + x_3 = 0 \quad \text{(modulo 2)}$$
$$x_2 + x_4 = 0 \quad \text{(modulo 2)}$$
$$x_1 + x_2 + x_3 + x_4 = 0 \quad \text{(modulo 2).}$$

Only two of these equations are needed, so, for example, a possible parity check matrix is given by

$$\mathrm{H} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

Activity 10

Find two other possible parity check matrices for Code 3.

Activity 11

Find a parity check matrix for Code 5 in Appendix 5.

9.4 Decoding using parity check matrices

Returning to the vector notation introduced earlier, suppose that the codeword **x** is transmitted, resulting in word

$$\mathbf{r} = \mathbf{x} + \mathbf{e}$$

being received. Hence **e** is the error word that has corrupted **x**.

For example, suppose the codeword transmitted is (1 1 1 0 0) but that the received word is $\mathbf{r} =$ (0 1 1 0 0). This means that the error word is given by

$$\mathbf{e} = (1\ 0\ 0\ 0\ 0).$$

Parity check matrices can be very useful for finding out the most likely errors in transmission in the case of **linear** codes.

A linear code, with parity check matrix H consists of all the words **x** which satisfy the equation

$$H\mathbf{x}' = 0.$$

Example

Show that Code 3 is linear.

Solution

A parity check matrix for Code 3 is given by

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \text{ (see Activity 10)}$$

Now the equation $H\mathbf{x}' = \mathbf{0}$ (modulo 2) can be written as

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

or
$$x_1 + x_3 = 0$$
$$x_2 + x_4 = 0.$$

If $x_1 = 1$, then $x_3 = 1$ (remember addition is modulo 2) whereas $x_1 = 0$ means $x_3 = 0$. Similarly for x_2 and x_4. This gives the following codewords

1 0 1 0

1 1 1 1

0 1 0 1

0 0 0 0

which is Code 3.

The importance of linear codes is that if **x** and **y** are two codewords in the linear code with parity check matrix H,

$$H(x+y)' = H(x'+y')$$

$$= Hx'+Hy'$$

$$= 0+0$$

$$= 0.$$

Hence $x+y$ is also a codeword.

The reverse is also true. That is, for every possible codeword **x** and **y**, if $x+y$ is also a codeword, then the code is linear.

Activity 12

Show that the code with codewords

0 0 0 0 0 0 0 0 0 0
0 1 1 0 1 1 0 0 1 0
1 0 0 1 0 0 1 1 0 1
1 1 1 1 1 1 1 1 1 1

is a linear code.

Now for a linear code, if $r = x + e$ is the received word, then

$$Hr' = H(x+e)'$$

$$= H(x'+e')$$

$$= Hx'+He'$$

$$= 0+He'$$

$$\Rightarrow \quad \boxed{Hr' = He'}$$

Note that this result shows that H**r**' is independent of the codeword transmitted.

Example

Suppose a codeword from Code 2 is received as
$\mathrm{r} = (1\,1\,0\,1\,1\,1)$. What is the most likely codeword sent?

Solution

$$\mathrm{H\,r} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

This result corresponds to the **last** column of H, so you would conclude that the codeword sent is in error in its last digit, and should have been

1 1 0 1 1 0.

Activity 13

For the following words

(a) 1 1 1 0 0 0 0 (b) 0 1 1 1 0 1 1

use a parity check matrix to determine which codewords from Code 5 were actually transmitted.

So the parity check matrix provides a means of finding the most likely error in transmission for linear codes.

9.5 Cyclic codes

Many codes have been designed to meet a variety of situations. One special sort of code is called a **cyclic** code.

Codes are called **cyclic** if they have the property that whenever

$$\mathrm{x} = (x_1\ x_2\ \dots\ x_n)$$

is a codeword, then so is x^* defined by

$$\mathrm{x}^* = (x_n\ x_1\, x_2\ \dots\ x_{n-2}\, x_{n-1}).$$

Example

Show that Code 1 in Appendix 5 is a cyclic code.

Solution

0 0 0 0 ← here $x = x^*$

1 1 0 0

1 0 1 0

1 0 0 1

0 1 1 0

0 1 0 1

0 0 1 1

1 1 1 1 ← here $x = x^*$

Hence the code is cyclic.

Activity 14

Show that Code 3 is a cyclic code.

Exercise 9B

1. Consider the linear code whose eight codewords are as follows:

0 0 1 1 1 0 1	0 1 0 1 0 1 1
0 1 1 0 1 1 0	1 0 0 0 1 1 1
1 0 1 1 0 1 0	1 1 0 1 1 0 0
1 1 1 0 0 0 1	0 0 0 0 0 0 0.

   (a) Find the distance between any two codewords, and hence find the minimum distance.

   (b) Find the number of errors in a transmitted codeword which can be detected and corrected by this code.

   (c) A codeword is transmitted and the binary word 1 0 0 1 1 0 0 is received. Which codeword is most likely to have been transmitted?

2. Let C be the code with the parity check matrix

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}.$$

   Find the codewords of C and write down the minimum distance of C.

3. The eight codewords of a linear code are as follows:

   0 0 0 0 0 0 0
   0 0 1 1 1 0 1
   1 0 0 1 0 1 1
   1 0 1 0 1 1 0
   1 1 0 0 1 0 1
   1 1 1 1 0 0 0
   0 1 1 0 0 1 1
   0 1 0 1 1 1 0

   (a) State the minimum distance of this code.

   (b) How many errors per received codeword can this code
   (i) correct (ii) detect?

   (c) A codeword is transmitted and the binary word 0 1 0 0 1 0 1 is received. Which of the eight codewords is most likely to have been the one transmitted?

4. Show that the code

0 0 0 0 0 0	1 0 1 0 1 1
0 0 0 1 1 1	1 0 1 1 0 0
0 1 1 0 0 1	1 1 0 0 1 0
0 1 1 1 1 0	1 1 0 1 0 1

   is linear and find a parity check matrix. Use it to decode the received message 0 1 0 1 1 0.

9.6 Miscellaneous Exercises

1. The code C has parity-check matrix

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

   (a) Write down the length, the dimension, and the rate of this code.

   (b) A codeword of C is transmited and incorrectly received as 0111000. Find the possible error and the transmitted codeword, assuming that only one error has occurred.

2. Consider the code whose codewords are

   0000000000, 0110110010, 1001001101,

   1111111111.

   (a) How many errors does this code **simultaneously** correct and detect?

   (b) If a message is received as 0110111111, which codeword is most likely to have been transmitted?

   (c) Is this code a linear code?

   (In each part, give reasons for your answer.)

3. One of the codewords of a cyclic code is 1001110.

   (a) List the other six words of the code.

   (b) What is the Hamming distance of this code?

   (c) How many errors in a codeword can be simultaneously detected and corrected? Give a brief reason for your answer.

   (d) Show that the code is not linear. What is the minimum number of codewords which need to be added to make this code linear?

   (e) The matrix

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

   is a parity check matrix for this code. Show how to use it to correct the received message

   0 1 1 0 1 1 1 1 1 1 0 1 0 0 0 0 1 1 1 1 1.

*4. Consider the linear code C whose eight codewords are as follows:

   0000000000, 1001011100,

   0100101110, 1101110010,

   0010010111, 1011001011,

   0110111001, 1111100101.

   (a) What is the minimum distance of this code? How many errors in a transmitted codeword can this code simultaneously correct and detect?

   (b) The matrix

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

   is a parity-check matrix for C. Use it to decode the received word 0010001111.