Coding Theory Homework

Week 1 (Section 1.1 - 1.4)

Exercise 1.2.1

> List all the words of length 3; of length 4; of length 5

Length 3	Length 4	Length 5
000	0000	00000
001	0001	00001
010	0010	00010
011	0011	00011
100	0100	00100
101	0101	00101
110	0110	00110
111	0111	00111
	1000	01000
	1001	01001
	1010	01010
	1011	01011
	1100	01100
	1101	01101
	1110	01110
	1111	01111
		10000
		10001
		10010
		10011
		10100
		10101
		10110
		10111
		11000

		11001
		11010
		11011
		11100
		11101
		11110
		11111

Exercise 1.2.2

> Find a formula for the total number of words of length *n*

2 ^ n

Length 3: 2 ^ 3 = 8
Length 4: 2 ^ 4 = 16
Length 5: 2 ^ 5 = 32

Exercise 1.2.3

> Let *C* be the code consisting of all the words of length 6 having an even number of ones. List the codewords in *C*.

```
000 000    000 011    000 101    000 110
001 001    001 010    001 100    001 111
010 001    010 010    010 100    010 111
011 000    011 011    011 101    011 110
100 001    100 010    100 100    100 111
101 000    101 011    101 101    101 110
110 000    110 011    110 101    110 110
111 001    111 010    111 100    111 111
```

Exercise 1.2.4

> Explain why a channel with $p = 0$ is uninteresting

A channel with $p = 1$ is a channel where no digit is sent wrongly and is thus correct. A channel with $p = 0$ is a channel where every digit is sent wrongly. The decoder can simply flip every digit to get the original message.

Exercise 1.2.5

> Explain how to convert a channel with $0 < p <= 1/2$ into a channel with $1/2 <= p < 1$

When a channel has a $p < 1/2$ more than half of the digits will be flipped during transmission. Using this knowledge we can have the decoder flip every digit it receives so that the majority of the digits will be correct.

Exercise 1.2.6

> What can be said about a channel with $p = 1/2$

Every single digit has a 50% chance to be flipped during the transmission.

Exercise 1.3.4

> Let *C* be the code of all words of length 3. Determine which codeword was most likely sent if 001 is received.

001

Exercise 1.3.5

> Add a parity check digit to the codewords in the code of Exercise 1.3.4, and use the resulting code *C* to answer the following questions
>
> \a. If 1101 is received can we detect an error?

Yes, the amount of ones in the received word is not even. By adding a partity check digit we made sure that every word in the code has an even amount of ones.

> \b. If 1101 is received what codewords were most likely to have been transmitted?

0101, 1001, 1111 or 1100.

> \c. Is any word of length 4 that is not in the code, closest to the unique codeword?

No, for every wordt of length 4 that is not in the code there are *four* closest codewords.

Exercise 1.3.6

> Repeat each codeword in the code *C* defined in Exercise 1.3.4 three times to form a repetition code of lenght 9. Find the closest codewords to the following received words

Received codeword	Closest codeword
001 000 001	001 001 001
011 001 011	011 001 011
101 000 101	101 101 101
100 000 010	000 000 000

Exercise 1.3.7

> Find the maximum number of codewords of length n = 4 in a code in which a single error can be detected.

```
0000    0011    0101    0110
1001    1010    1100    1111
```

Exercise 1.3.8

> Repeat exercise 1.3.7 for n = 5, n = 6 and for general *n*

2 ^ n-1

2 ^ (4 - 1) = 8

2 ^ (5 - 1) = 16
2 ^ (6 - 1) = 32

Exercise 1.4.1

> Find the information rate for each of the codes in Exercise 1.3.4, 1.3.5 and 1.3.6

1.3.4: Information rate of 1
1.3.5: Information rate of 3/4
1.3.6: Information rate of 1/3