Lecture 4: Source coding

Exercise 4.1:
In a Hamming code word a check bit toggles. Can the mistake be detected and corrected and if yes, how?

Exercise 4.2:
The following Hamming code word is given: 01111001111. Create an error with as few changes as possible that can not be detected. How do you know that less changes of bits can not pass undetected?

Exercise 4.3:
A number of e bit errors should be corrected. Explain why a distance of 2e is not sufficient for the code?
Lecture 4: Source coding

Exercise 4.4:

In the last lecture we have seen an estimate of how many redundant bits are necessary to detect and correct 1 bit error. Now do the same estimation for 2 bit errors. It is not necessary to find a particular code, only a lower bound for the number of check bits is of interest.

How many bits are necessary to protect a 7 bits ASCII code against at most 2 toggled bits?

Exercise 4.5:

Explain why protecting n bits of data against single bit errors with forward error correction requires only $O(\log(n))$ check bits while protecting a fixed number of c bits against a number of n toggled bits ($n \ll c$) requires $O(2^n)$ check bits. Hint: Think of the table consisting of valid and invalid code words and of the method in which bits a checked in the Hamming code.
Exercise Data Networks

Exercise 4.6:

A sender wants to transmit the following 32 bits

\[ 10110100 \ 01011011 \ 01010101 \ 10110110 \]

in four chunk-packets, each of which contains 8 bits. Both, sender and receiver use the same random number generator which produces the following bit-stream:

\[ 1110 \ 0101 \ 1001 \ 0110 \]

For data transmission, the Random Linear Fountain Code from the lecture is used.
Exercise 4.6:

**Procedure:**
- Divide the message into chunks.
  - Combine the chunks bit-wise according to the bit-merging Sender vector which is taken from the output of the random number Side generator.

**Receiver Side**
- Collect the incoming chunks until a sufficient number is received.
  - Sufficient means, that their merging vectors are linearly independent. The merging vectors are taken from the output of the random number generator as was done on the sender side.
  - After having gathered enough data, calculate the modulo 2 inverse matrix of the matrix formed by the merging vectors.
  - XOR the received chunks according to the inverse matrix.