Please discuss the following problems among the students in your group. Some of the groups will be selected to present sample solutions at the group presentation on Wednesday.

Day Two

Problem 1

Consider the linear code $\mathcal{C}$ of length 10 and dimension 9 over $\mathbb{F}_{11}$ whose parity-check matrix is given by

$$\mathbf{H} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix}.$$

(Note that $\mathbb{F}_{11}$ is a fancy way of saying that one is computing with integers where addition and multiplication are modulo 11.)

This code is used for ISBNs (International Standard Book Numbers). Actually, in that case, the first nine symbols (the information symbols) are confined to lie in $\{0, 1, 2, \ldots, 9\}$, whereas the last symbol (check symbol) lies in $\{0, 1, 2, \ldots, 9, \mathrm{X}=10\}$.

(a) Wikipedia says that the parity-check matrix for the ISBN code is

$$\mathbf{H} = \begin{pmatrix} 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}.$$

Why is there no contradiction with the above definition?

(b) Show that $\mathcal{C}$ has minimum (Hamming) distance two.

(c) Show that there is a decoder for $\mathcal{C}$ which can detect one single error or the transposition of two codeword positions.

Remark: Note that the above is the description of the ISBN-10 standard, which uses codewords of length 10. There is also the newer ISBN-13 standard, which uses 13 symbols, a different parity-check matrix, and modulo-10 arithmetic.

Problem 2

Consider a binary linear code $\mathcal{C}$ of length n and dimension k which is described by a parity-check matrix $\mathbf{H}$ of size $m \times n$. (Often, $m = n - k$, but $m > n - k$ is possible when $\mathbf{H}$ has some linearly dependent parity checks.)

The Tanner graph T associated with $\mathbf{H}$ is a bipartite graph with n bit nodes and m check nodes, whereby the i-th bit node is connected to the j-th check node via an edge if the entry in the j-th row and the i-th column of $\mathbf{H}$ equals 1. As we have seen, the Tanner graph T is a useful tool for visualizing various types of decoders.

In graph theory, the degree of a vertex is defined to be the number of edges incident on that vertex. In terms of T, this means the following:

- the degree of a bit node equals the number of parity checks this bit is involved in;
- the degree of a check node equals the number of bits in the corresponding parity-check equation.

The bit and check node degree distributions *from the node perspective*, i.e.,

$$\Lambda(x) = \sum_{\ell} \Lambda_\ell x^\ell \quad \text{and} \quad \mathrm{P}(x) = \sum_{\ell} \mathrm{P}_\ell x^\ell,$$

of T give us the following information about T:

- Λ_ℓ is the fraction of bit nodes that have degree ℓ;
- P_ℓ is the fraction of check nodes that have degree ℓ.
  (Here, "P" is the Greek capital letter "rho".)

These degree distributions are useful for constructing codes. However, when analyzing decoders, one often needs the bit and check node degree distributions *from the edge perspective*, i.e.,

$$\lambda(x) = \sum_{\ell} \lambda_\ell x^{\ell-1} \quad \text{and} \quad \rho(x) = \sum_{\ell} \rho_\ell x^{\ell-1}.$$

(Note the "-1" in the exponents.)

These degree distributions give the following information about T:

- λ_ℓ is the fraction of edges that are incident on a bit node of degree ℓ;
- ρ_ℓ is the fraction of edges that are incident on a check node of degree ℓ.

(a) Express the parameters $(\lambda_\ell)_\ell$ and $(\rho_\ell)_\ell$ in terms of the parameters $(\Lambda_\ell)_\ell$ and $(\mathrm{P}_\ell)_\ell$, and vice-versa.

(b) Express $\lambda(x)$ and $\rho(x)$ in terms of $\Lambda(x)$ and $\mathrm{P}(x)$, and vice-versa.
Hint: try polynomial evaluation and/or differentiation and/or integration.

Problem 3

Consider the transmission of a binary codeword of length n over a binary symmetric channel (BSC) with cross-over probability ε, where $0 \leq \varepsilon \leq 1$. (A BSC with cross-over probability ε is a channel that flips a transmitted bit with probability ε.)

Let Z_i, $i = 1, \ldots, n$, be independent discrete random variables that tell us if a bit flip happened at time i, i.e., when a bit flip happened at time i then $z_i = 1$, otherwise $z_i = 0$. The probability mass function of Z_i is therefore $P_{Z_i}(0) = 1 - \varepsilon$ and $P_{Z_i}(1) = \varepsilon$. Furthermore, let

$$Z = Z_1 + \cdots + Z_n \quad (\text{in } \mathbb{Z})$$

be the total number of bit flips and let

$$Z' = Z/n$$

be the relative number of bit flips.

(a) What are $\mathrm{E}[Z_i]$ and $\mathrm{Var}[Z_i]$ for $i = 1, \ldots, n$?

(b) What are $\mathrm{E}[Z]$ and $\mathrm{Var}[Z]$?

(c) What are $\mathrm{E}[Z']$ and $\mathrm{Var}[Z']$?

(d) Let $\varepsilon = 0.1$. For $n = 10$, $n = 1000$, and $n = 100000$, compute the the mean $\mathrm{E}[Z']$ and standard deviation $\sqrt{\mathrm{Var}[Z']}$.

(e) The smaller the standard deviation (or the variance) of a random variable is, the more can we expect that the value taken on in a random experiment will be near the expected value. What does this and the above calculations imply about the relative number of errors when using longer and longer block lengths n?

Calvin, the coding theorist:

Problem 4

The following is a SUDOKU puzzle:

		6		1	8		4	2
7		9				8	5	
8	4	2		6		3		
	9		3					
	2		8		1		7	
					4		3	
		8		4		5	9	1
	7	1				4		3
4	5		1	8		2		

As you certainly know, the task is to fill this 9×9 array with the numbers 1 to 9 such that

- in every row, the numbers 1 to 9 appear exactly once;
- in every column, the numbers 1 to 9 appear exactly once;
- in every 3×3 sub-array, the numbers 1 to 9 appear exactly once.

The following represents a filled-out SUDOKU puzzle:

1	2	5	3	9	6	8	7	4
4	6	3	8	7	5	2	9	1
7	9	8	2	4	1	5	3	6
5	4	7	6	1	2	9	8	3
2	3	9	5	8	4	1	6	7
8	1	6	9	3	7	4	2	5
6	8	1	7	5	9	3	4	2
3	7	4	1	2	8	6	5	9
9	5	2	4	6	3	7	1	8

(a) Can you draw a Tanner graph for the SUDOKU puzzle? I.e., in the same manner in which a Tanner graph represents what constraints the symbols of a codeword have to satisfy, can you come up with a Tanner graph that represents a correctly filled-out SUDOKU puzzle array? In particular: How many symbol nodes will this Tanner graph have? What alphabet do these symbol nodes have? How many check nodes will this Tanner graph have? What checks do these check nodes impose?

(b) There are some simple heuristics for solving a SUDOKU puzzle. For example, if all entries but one in a row are filled out, it is easy to find the value of the remaining entry. Can you express this, and other similar simple heuristics, as sending suitable information ("messages") along the edges of the Tanner graph and doing processing of the messages at the nodes?