MTAT.05.082: Introduction to Coding Theory **University of Tartu**

Homework Assignment 2

Due date: November 2, 2018

It is possible to collect up to 110 points in this homework assignment.

Problem 1. 1. Let $\mathcal{C}$ be an (n, M, d) code over the field $\mathbb{F}$ with $d \geq 2$. Fix some arbitrary i, $1 \leq i \leq n$. Define *the code punctured at coordinate* i as

$$\mathcal{C}' = \{ (c_1 c_2 \cdots c_{i-1} c_{i+1} c_{i+2} \cdots c_n) \mid (c_1 c_2 \cdots c_n) \in \mathcal{C} \} .$$

Prove that $\mathcal{C}'$ is an $(n-1, M, d')$ code over $\mathbb{F}$, where $d' \geq d - 1$.

2. Generalize the definition of the puncturing by defining the code $\mathcal{C}''$ punctured at $r \geq 1$ coordinates $i_1, i_2, \cdots, i_r$, where $1 \leq i_1 < i_2 < \cdots < i_r \leq n$.

3. Show that for $1 \leq r \leq d-1$, the code $\mathcal{C}''$ defined in (b) is an $(n-r, M, d'')$ code, where $d'' \geq d - r$.

4. Assume that the (n, M, d) code $\mathcal{C}$ over $\mathbb{F}$ is used to transmit information, and that there are ρ erasures and τ errors in the received word. Show that there exists a decoder that recovers the original codeword, whenever $\rho + 2\tau \leq d - 1$.
Hint: consider the code punctured in ρ erased coordinates. Use the minimum distance of that code.

Problem 2. For each pair of the polynomials $a(x)$ and $b(x)$ over finite field $\mathbb{F}$:

(i) find $\gcd(a(x), b(x))$ by using Euclid's algorithm over $\mathbb{F}$;

(ii) Find polynomials $s(x)$ and $t(x)$ over $\mathbb{F}$ such that

$$s(x) \cdot a(x) + t(x) \cdot b(x) = \gcd(a(x), b(x)) ,$$

when

1. $a(x) = x^4 + x$, $b(x) = x^3 + x^2$, and $\mathbb{F} = \mathbb{F}_2$ (field of integer residues modulo 2).

2. $a(x) = x^3 + 4x^2 + 3x$, $b(x) = x^3 + 3x^2 + 4x + 2$, and $\mathbb{F} = \mathbb{F}_5$ (field of integer residues modulo 5).

Problem 3. 1. Let a be a non-zero element in the finite field $\mathbb{F}$ with q elements, and consider the mapping $f : \mathbb{F} \to \mathbb{F}$ defined by $f(x) = a \cdot x$. Show that each element of $\mathbb{F}$ is the image under f of exactly one $x \in \mathbb{F}$.

2. Now, $(a_1, a_2, \cdots, a_k)$ is a non-zero vector over $\mathbb{F}$, and the mapping $g : \mathbb{F}^k \to \mathbb{F}$ is defined by $g(x_1, x_2, \cdots, x_k) = \sum_{i=1}^{k} a_i x_i$. Show, that each element of $\mathbb{F}$ is the image under g of exactly q^{k-1} vectors in $\mathbb{F}^k$.

3. Let $\mathcal{C}$ be a linear $[n, k, d]$ code over $\mathbb{F}_q$ and let T be a $q^k \times n$ array whose rows are the codewords of $\mathcal{C}$. Show that each element of $\mathbb{F}_q$ appears in every nonzero column in T exactly q^{k-1} times.

   Hint: use 2. You could use it as granted even if you have not proved it.

Problem 4. Programming task, part 1: polynomials over Galois fields.

During the course you will develop a small coding library. We do not limit you in choice of the programming language or paradigm. However, you are supposed to use **only core language** and perhaps some standard general-purpose library which is a part of the language – for example, STL for C++, etc. By no mean you can use any libraries or frameworks that provide ready-made solutions for finite fields, polynomials, algorithms we study, and linear codes. Possible languages include

Pascal/C/C++/Java/C#/Python/Haskell.

If you want to use some other languages, the full list of languages we can provide is here: `https://www.hackerrank.com/environment`. Pay attention for time and memory limitations. Although we will not usually put "heavy" tests which normally come close to the limits. This is just to make sure you indeed implement efficient algorithms (instead of, say, brute force exhaustive checking for decoding algorithms).

If you really want to use some other programming language, please contact the teaching staff.

Your program will be tested automatically in HackerRank environment (`https://www.hackerrank.com`). If you do not have an account, go there and create one. Please, use your automatically generated pseudonym from the course page (see the last column here `https://courses.cs.ut.ee/user/my_data`). You can change your Hackerrank username in your profile: `https://www.hackerrank.com/settings/account`.

Each programming assignment will be represented by a "contest" at HackerRank, which in turn will consist of several "challenges", i.e. separate tasks.

The contest for this assignment is here: `www.hackerrank.com/coding-theory-a18-hw2`. Go there, sign up, and start solving the challenges. The problems have some sample test cases, so that you can understand better what is expected from your programme.

The deadline for problems submission is the same as for the whole homework assignment! However, it does not matter how many times you try to submit your programs before the deadline. Only the best result will be counted.

In brief, you will need to implement the following operations with polynomials over simple Galois fields: addition, subtraction, multiplication, division with remainder, and calculation of the greatest common divisor (GCD). Each of this operations has a dedicated challenge, but be wise and do not write the same code again and again. It is a good idea to organise your code with some structures provided by the language of your choice (functions, classes, ADT, etc.) We will use these operations with polynomials in further programming assignments, so write your code smart.

This is the first time we use HackerRank system, so there potentially could be some technical issues. Please let the teaching stuff know immediately if you have any problems or questions.