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

Homework Assignment 6

Due date: December 24, 2018

---

1. In this question, you will use your previously written code for a small GRS-decoding library. The task will be automatically tested on different inputs. One run of your program should do the following:

   - read the input from the console, which describes one instance of the problem of decoding a GRS code over a simple finite field $\mathbb{F}_p$;
   - decode the received word;
   - write output to console

   Detailed description of decoding problem. You are given a prime field $\mathbb{F}_p$, a parity-check matrix H of an $[n, k, d]$ GRS code (described by its $\alpha_1, \alpha_2, \ldots, \alpha_n$ and $v_1, v_2, \ldots, v_n$), and a received word $\bar{\boldsymbol{y}} = (y_1, y_2, \ldots, y_n)$. Your task is to find the original codeword $\bar{\boldsymbol{c}} = (c_1, c_2, \ldots, c_n)$.

   Input and output specification.

   The input has the following format:

   (a) The first line contains one prime number p which defines the field $\mathbb{F}_p$.

   (b) The second line contains two integer numbers n and k, parameters of the GRS code.

   (c) The third line contains n numbers separated by the spaces: $\alpha_1, \alpha_2, \ldots, \alpha_n$. Each number is a non-zero element of $\mathbb{F}_p$ represented as an integer in the range $[1, p-1]$.

   (d) The fourth line contains n numbers separated by the spaces: $v_1, v_2, \ldots, v_n$. Each number is a non-zero element of $\mathbb{F}_p$ represented as an integer in the range $[1, p-1]$.

   (e) The fifth line contains n numbers separated by the spaces: $y_1, y_2, \ldots, y_n$. Each number is an element of $\mathbb{F}_p$ represented as an integer in the range $[0, p-1]$.

   The parity-check matrix of the code has the following form:

$$H = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \alpha_3 & \ldots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \alpha_3^2 & \ldots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{n-k-1} & \alpha_2^{n-k-1} & \alpha_3^{n-k-1} & \ldots & \alpha_n^{n-k-1} \end{pmatrix} \cdot \begin{pmatrix} v_1 & 0 & \ldots & 0 \\ 0 & v_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & v_n \end{pmatrix}$$

   The output should have the only line that contains n elements $c_1, c_2, \ldots, c_n$. Each element should be represented as integer in range $[0, p-1]$.

   The contest for this assignment is here:

https://www.hackerrank.com/coding-theory-a18-hw6

It has only one challenge. Go there and start solving. The problem 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.

Example of input:

```
7
6 2
1 2 3 4 5 6
1 1 1 1 1 1
1 2 5 4 5 1
```

Expected output:

```
1 2 3 4 5 6
```

Explanation. The input above describes the following problem. You are given the field $\mathbb{F}_7$ and the $[6, 2]$ GRS code with $\alpha_1 = 1$, $\alpha_2 = 2$, $\alpha_3 = 3$, $\alpha_4 = 4$, $\alpha_5 = 5$, and $\alpha_6 = 6$, as well as $v_1 = v_2 = v_3 = v_4 = v_5 = v_6 = 1$. The received word is $\bar{\boldsymbol{y}} = (1, 2, 5, 4, 5, 1)$.

Decoding of $\bar{\boldsymbol{y}}$ gives original codeword $\bar{\boldsymbol{c}} = (1, 2, 3, 4, 5, 6)$ which is output.

Example 2 of input:

```
29
20 7
19 5 4 6 10 3 16 27 2 15 26 28 23 14 20 25 22 18 24 13
9 25 21 3 8 18 17 23 4 28 20 2 27 15 22 11 14 13 24 16
19 10 17 17 9 20 1 8 22 18 14 21 12 17 15 25 7 22 6 5
```

Expected output:

```
19 18 9 9 9 20 1 8 22 18 14 21 12 17 15 25 7 22 6 5
```

Explanation. The input above describes the following problem. You are given the field $\mathbb{F}_{29}$ and the $[20, 7]$ GRS code with

$$\bar{\boldsymbol{\alpha}} = (19, 5, 4, 6, 10, 3, 16, 27, 2, 15, 26, 28, 23, 14, 20, 25, 22, 18, 24, 13) \ ,$$

and

$$\bar{\boldsymbol{v}} = (9, 25, 21, 3, 8, 18, 17, 23, 4, 28, 20, 2, 27, 15, 22, 11, 14, 13, 24, 16) \ .$$

The received word is

$$\bar{\boldsymbol{y}} = (19, 10, 17, 17, 9, 20, 1, 8, 22, 18, 14, 21, 12, 17, 15, 25, 7, 22, 6, 5) \ .$$

After the decoding, we obtain the original codeword

$$\bar{\boldsymbol{c}} = (19, 18, 9, 9, 9, 20, 1, 8, 22, 18, 14, 21, 12, 17, 15, 25, 7, 22, 6, 5) \ .$$

Notes:

- You are free to choose any of the decoding methods described during the course.
- All the input data is assumed to be correct so that you do not need to check the correctness of the input (e.g. that p is prime, $k < n$, $\bar{\boldsymbol{y}}$ contains not more than $\tau = \lfloor (d-1)/2 \rfloor$ errors, etc.)
- We consider only prime (i.e. not extended) fields.
- You are not allowed to use any libraries except for the core of a programming language you use. However, you are allowed to use you own code. If unsure, please contact the teaching staff to confirm if some functionality is allowed for use.
- Efficiency is again not a serious issue, hence you should not care much about it. However, note that the size of the input is long enough to ensure that brute force methods will not work. On the other hand, any of the decoding methods from the course have a complexity small enough for the test cases. For example, our experimental "quick and dirty" Python script is able to decode a codes of length $n = 500$ in not more than one second.