## Homework #10

*Due Tue, April 21st by 23:59pm in filedrop*

### Reading and plan for the next week:

1. For this homework assignment read 7.5, 6.1 and 6.2 in Lindell's notes and class notes from Lectures 19-22.

2. Plan for next week: Asymptotically good codes. This material is not in our main book, but it is discussed in Lindell's notes (see 4.1, 4.2 and Chapter 7). There is way too much material in those sections, so I will probably just discuss the main results and ideas, skipping many proofs.

### Problems:

**1.** Let $C$ be a narrow-sense RS (Reed-Solomon) code over $\mathbb{F}_7$ (as defined in Lecture 20) of dimension $k = 4$ (such a code is unique up to the choice of primitive element $\alpha$). Write down explicitly a GM for $C$, a PCM for $C$ and the generator polynomial for $C$ (all matrix entries and coefficients of the polynomial in your answer should be explicit elements of $\mathbb{F}_7$).

**2.** Let $F$ be a finite field, $q = |F|$ and $n = q - 1$.

(a) Given $\beta \in F$, let $\Sigma(\beta) = \sum\limits_{j=0}^{n-1} \beta^j$. Prove that $\Sigma(1) = -1$ and $\Sigma(\beta) = 0$ for $\beta \neq 0, 1$. **Hint:** Consider the cases $\beta = 0, 1$ separately. For all other $\beta$ recall how to find the sum of a finite geometric progression.

(b) Let $\alpha \in F$ be a primitive element, set $\alpha_i = \alpha^{i-1}$ for $1 \leq i \leq n$, and let $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$ and $\vec{v} = (v_1, \ldots, v_n)$ for some nonzero $v_i \in F$. Let $C = GRS(\vec{\alpha}, \vec{v})$ for some $k$. Recall from Lecture 20 that GRS codes of this type are called primitive.

For each $1 \leq i \leq n$ set $w_i = \frac{\alpha_i}{v_i} = \frac{\alpha^{i-1}}{v_i}$. Use (a) to prove that these elements $w_1, \ldots, w_n$ are valid GM multipliers for $C$, that is, they satisfy the conclusion of Theorem 19.4 from class.

**3.** Give a new proof of the fact that GRS codes are MDS using polynomial representation of GRS codes (see the section "Explicit description of elements of $GRS_k(\vec{\alpha}, \vec{v})$" at the end of Lecture 19). You only need to use the description obtained in this section, not the original definition.

**Hint:** use the fact that for any field $F$, a nonzero polynomial of degree $m$ over $F$ cannot have more than $m$ roots.

**4.** Prove Lemma 21.2 from class (in the proof make sure to use the definition given in Lecture 21). **Hint:** for the second part of Lemma 21.2 (dealing with burst error correction) it may be convenient to "negate the definition of an $E$-error correcting code", that is, to rewrite the definition in the form "A code $C$ is NOT $E$-error correcting $\iff$ ..."

**5.** Prove Theorem 22.1 from class. Recall that in class we discussed a specific example illustrating the proof. As in the example, it is probably most convenient to use part of Lemma 21.2 recalled at the beginning of Lecture 22.

**6.** Let $D$ be the binary cyclic code of length 9 with generator polynomial $g(x) = 1 + x^3 + x^6$.

(a) Show that $D$ has the form $IL(C^m)$ (in the notations from Lecture 22) for some (specific) $C$ and $m$. Deduce that $D$ is 3-burst error correcting.

(b) Use the error trapping algorithm to decode the word $w = 110111000$ (as a polynomial $w = 1 + x + x^3 + x^4 + x^5$).

Some remarks on 6(b):

(i) Using the description of $D$ from (a), it is easy to guess the answer (and once you correctly guessed $c$, you can justify that you guessed correctly using the fact that $D$ is 3-burst error correcting). Nevertheless, you should still show how to get the answer using error trapping.

(ii) The syndromes $s_i = S(x^i \odot w)$ are not hard to compute directly from definition, but you may use the recursive formula $s_{i+1} = (xs_i) \mod g(x)$ (if you do use this formula, at least think why it should be true).

(iii) HW# 9.1 describes a "shortcut" for computing the remainders modulo $x^n - 1$. There is a similar shortcut for finding the remainders modulo any polynomial (try to formulate it in general). The amount of time it saves depends on the number of nonzero monomials in the poynomial you are dividing by (the fewer monomials, the better). So, for $g(x) = x^6 + x^3 + 1$ which has 3 monomials, it would not work as well as for $x^n - 1$, but still pretty well.