Solutions 10

Exercise 10.1.

We must find polynomials $g(x)$ and $h(x)$ such that $g(x_i) - y_i h(x_i) = 0$ for all i, and furthermore, $\deg(g) < 4$ and $\deg(h) \leq 1$. $h(x)$ will be the error locator polynomial. We thus want to find a solution to the system

$$A \begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ h_0 \\ h_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

where

$$A = \begin{pmatrix} 1 & 1 & 1^2 & 1^3 & -5 & -5 \cdot 1 \\ 1 & 2 & 2^2 & 2^3 & -2 & -2 \cdot 2 \\ 1 & 3 & 3^2 & 3^3 & -6 & -6 \cdot 3 \\ 1 & 4 & 4^2 & 4^3 & -3 & -3 \cdot 4 \\ 1 & 5 & 5^2 & 5^3 & -5 & -5 \cdot 5 \end{pmatrix}.$$

Performing Gaussian elimination over $\mathbb{F}_7$ on the matrix A, we get the equivalent system

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 2 & 2 \\ 0 & 1 & 3 & 0 & 3 & 1 \\ 0 & 0 & 2 & 5 & 0 & 6 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 \end{pmatrix} \begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ h_0 \\ h_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Setting $h_1 = 1$, we can solve for h_0 and get $h_0 = 2$. Thus $h(x) = x - 2$, so that we know that the error occured at the position corresponding to the evaluation point 2. We discard the corresponding position and perform erasure decoding on the rest of the vector y. Note that a generator matrix for our code is

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2^2 & 3^2 & 4^2 & 5^2 \\ 1 & 2^3 & 3^3 & 4^3 & 5^2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 4 & 2 & 2 & 4 \\ 1 & 1 & 6 & 1 & 6 \end{pmatrix},$$

so that the codeword $(5, x, 6, 3, 5)$ that we are looking for is of the form

$$a(1,1,1,1,1) + b(1,2,3,4,5) + c(1,4,2,2,4) + d(1,1,6,1,6).$$

Solving the system

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 3 & 2 & 6 \\ 1 & 4 & 2 & 1 \\ 1 & 5 & 4 & 6 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ 3 \\ 5 \end{pmatrix},$$

we get $(a, b, c, d) = (4, 1, 2, 5)$ and hence the missing coordinate is $x = a + 2b + 4c + d = 5$, so that the sent codeword is $(5, 5, 6, 3, 5)$.

Exercise 10.2.

1. As a RS-code, $\mathcal{C}$ is MDS so $d = 15 - 6 + 1 = 10$. Balls up to radius 4 do not intersect so one can correct up to 4 errors by maximum likelihood. On the other hand, the W.-B. decoder can decode up to $\frac{n-k}{2} = \frac{15-6}{2} = 4.5$ errors, i.e. 4 errors.

2. For each (γ_i, r_i) with $0 \leq i < n$, we want to ensure that $p(x + \gamma_i, y + r_i)$ contains no monome $x^\alpha y^\beta$ of degree $\alpha+\gamma \leq r$: that gives $m(m+1)/2$ equations and $nm(m+1)/2 = 45$. On the other hand, the space of polynomials in $\mathbb{F}_{16}[x, y]$ of $(1, 5)$-degree $\leq \delta$ has dimension

$$\sum_{i=0}^{\lfloor \frac{\delta}{k-1} \rfloor} \sum_{j=0}^{\delta - i(k-1)} 1 = \sum_{i=0}^{\lfloor \frac{\delta}{5} \rfloor} (\delta + 1 - 5i) = (\delta + 1) \left(\left\lfloor \frac{\delta}{5} \right\rfloor + 1 \right) - \frac{5}{2} \left\lfloor \frac{\delta}{5} \right\rfloor \left(\left\lfloor \frac{\delta}{5} \right\rfloor + 1 \right)$$

   For $\delta = 17$, this evaluates to 42 and for $\delta = 18$ to 46. One should thus choose $\delta = 18$.

   Let $f \in \mathbb{F}_{16}[x]_{<6}$ be a polynomial such that $f(\gamma_i) = r_i$ for at least $\lceil \frac{\delta+1}{m} \rceil = 10$ indices i (In other word, the Hamming distance between f and the received codeword is ≤ 5). Then, $p(x, f(x))$ vanishes 10 times with multiplicity 2, so $p(x, f(x))$ is zero. In other words, $y = f(x)$ is a root of p viewed as a polynomial in y over the fraction field of rational functions in x. Thus, by division algorithm, $y - f(x)$ is a factor of this polynomial. This shows that any codeword with distance ≤ 5 to r will be obtained by this method, so 5 errors can be decoded.

3. We can now correct up to 6 errors.

Exercise 10.3.

1. Any (n, M, d) code is $\left(\frac{d-1}{2}, 1 \right)$-list decodable and vice-versa. This follows immediately from the definition of minimum distance.

2. First note that
$$l \leq \frac{n(t - (k-1))}{t^2 - (k-1)n} \leq n^2, \text{ if } t^2 > (k-1)n.$$
   Using the fact that $k - 1 = n - d$ and letting $e = n - t$, we see that there are at most n^2 codewords in a ball of radius e around any vector y in the space, provided $(n - e)^2 > n(n - d)$. But this is verified for $e < n - \sqrt{n(n-d)}$.

   This result is particularly useful when the minimum distance is relatively large (k not too large) with respect to n: it says that we can list-decode for a large number of errors and with a relatively short (polynomial in n) list of codewords.

   Note also that we have stated this bound for RS codes, but it is actually valid for any code, not necessarily linear (using d instead of $k - 1$), so that the list-decoding bound shown here applies to any code.

Exercise 10.4.

1. By assumption, we have that $Q(x, f(x)) \equiv 0$. Thus we must have $Q(0, f(0)) = 0$. But this is equal to $A_0(\beta)$ by the expansion of $Q(x, y)$ as a polynomial in $\mathbb{F}_q[y][x]$.

2. First, we write

$$(y - f_0 - f_1 x - \cdots - f_{k-1}x^{k-1})(\psi_0(y) + \psi_1(y)x + \cdots) = A_0(y) + A_1(y)x + \cdots,$$

and then equate the coefficient of x^0 on both sides, to obtain

$$(y - f_0)\psi_0(y) = A_0(y) \Rightarrow \psi_0(y) = A_0(y)/(y - \beta).$$

Similarly, equating the coefficient of x^1 on both sides yields

$$A_1(y) = (y - \beta)\psi_1(y) - f_1\psi_0(y).$$

Instantiating the latter identity with $y = \beta$ gives $f_1 = -A_1(\beta)/\psi_0(\beta)$, as desired.

3. Similar to above, we equate the coefficient of x^i on both sides to obtain

$$\psi_i(y)(y - \beta) = A_i(y) + f_i\psi_0(y) + \cdots + f_1\psi_{i-1}(y),$$

which gives the identity

$$\psi_i(y) = \frac{A_i(y) + f_i\psi_0(y) + \cdots + f_1\psi_{i-1}(y)}{y - \beta},$$

and then instantiate with $y = \beta$ to obtain

$$f_i = -\frac{A_i(\beta) + f_1\psi_{i-1}(\beta) + \cdots + f_{i-1}\psi_1(\beta)}{\psi_0(\beta)}.$$

Note that these expressions are well-defined. Since we have asummed that β is a simple root of $A_0(y)$, it cannot be a root of $\psi_0(y) = \frac{A_0(y)}{y-\beta}$, so that we can divide by $\psi_0(\beta)$ in the expression for f_i. And in the expression for $\psi_i(y)$, note that $y - \beta$ does divide the numerator. Indeed,

$$\begin{aligned}
&A_i(\beta) + f_1\psi_{i-1}(\beta) + \cdots + f_{i-1}\psi_1(\beta) \\
&\quad = A_i(\beta) - \frac{A_i(\beta) + f_1\psi_{i-1}(\beta) + \cdots + f_{i-1}\psi_1(\beta)}{\psi_0(\beta)}\psi_{i-1}(\beta) + \cdots + f_{i-1}\psi_1(\beta) \\
&\quad = 0.
\end{aligned}$$

Then the algorithm would find coefficients $f_0, \ldots, f_{k-1}$ one by one using this recursion.

4. For the given $Q(x, y)$, we have

$$\begin{aligned}
A_0(y) &= y^5 + y^4 + y^3 + y \\
A_1(y) &= y^2 + y \\
A_2(y) &= y^3 + y^2 + 1 \\
A_3(y) &= y^4 + y^2 + y + 1 \\
A_4(y) &= y^3 \\
A_5(y) &= y^3 \\
A_6(y) &= 0 \\
A_7(y) &= 1.
\end{aligned}$$

We note that $A_0(y)$ has two simple roots: $\beta = 0$ and $\beta = 1$. First we set $\beta = 0$ and follows the recursions, which give

$$\begin{aligned}
f_0 &= \beta = 0 \\
\psi_0(y) &= y^4 + y^3 + y^2 + 1 \\
\psi_0(\beta) &= 1 \\
f_1 &= -A_1(\beta) = 0 \\
\psi_1(y) &= (A_1(y) + f_1\psi_0(y))/(y - \beta) = (y^2 + y)/y = y + 1 \\
f_2 &= -(A_2(\beta) + f_1\psi_1(\beta)) = 1 \\
\psi_2(y) &= (A_2(y) + f_2\psi_0(y) + f_1\psi_1(y))/(y - \beta) = (y^3 + y^2 + 1 + y^4 + y^3 + y^2 + 1)/y = y^3 \\
f_3 &= -(A_3(\beta) + f_1\psi_2(\beta) + f_2\psi_1(\beta)) = 0,
\end{aligned}$$

and we obtain a factor $y + x^2$. Then we set $\beta = 1$ and perform a simiar computation:

$$\begin{aligned}
f_0 &= \beta = 1 \\
\psi_0(y) &= A_0(y)/(y - \beta) = (y^5 + y^4 + y^3 + y)/(y - 1) = y^4 + y^2 + 1 \\
\psi_0(\beta) &= 1 \\
f_1 &= -A_1(\beta) = 0 \\
\psi_1(y) &= (A_1(y) + f_1\psi_0(y))/(y - \beta) = (y^2 + y)/(y - 1) = y \\
f_2 &= -(A_2(\beta) + f_1\psi_1(\beta)) = 1 \\
f_3 &= -(A_3(\beta) + f_1\psi_2(\beta) + f_2\psi_1(\beta)) = \psi_1(1) = 1,
\end{aligned}$$

and we obtain another factor $y + x^3 + x^2 + 1$.