EE/Ma 127b Error-Correcting Codes — Anxiao (Andrew) Jiang
*Draft of February 17, 2001* — 311 Moore

Homework Assignment 3, Solutions

**Problem 1.**
Proof:

$$\big(P(\alpha_0),\quad P(\alpha_1),\quad \cdots,\quad P(\alpha_{n-1}),\quad P(\infty)\big) = \big(I_0,\quad I_1,\quad \cdots,\quad I_{k-1}\big)\begin{pmatrix} 1 & 1 & \cdots & 1 & 0 \\ \alpha_0 & \alpha_1 & \cdots & \alpha_{n-1} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_0^{k-2} & \alpha_1^{k-2} & \cdots & \alpha_{n-1}^{k-2} & 0 \\ \alpha_0^{k-1} & \alpha_1^{k-1} & \cdots & \alpha_{n-1}^{k-1} & 1 \end{pmatrix}$$

is a linear transformation. So this is an $(n+1,k)$ linear code over F.

Below we study the minimum (nonzero) weight of this code.

(1) If $I_{k-1}=0$, $P(x)$ has degree less than $k-1$ and therefore has no more than $k-2$ roots. So among $P(\alpha_0),P(\alpha_1),\cdots,P(\alpha_{n-1})$ at most $k-2$ of them are zeroes. $\therefore$ The weight of the codeword is at least $n-(k-2)=n-k+2$.

(2) If $I_{k-1}\neq 0$, $P(x)$ has degree $k-1$ and therefore has no more than $k-1$ different roots. So among $P(\alpha_0),P(\alpha_1),\cdots,P(\alpha_{n-1})$ at most $k-1$ of them are zeroes. And we have $P(\infty)=I_{k-1}\neq 0$. $\therefore$ The weight of the codeword is at least $n-(k-1)+1=n-k+2$.

So the code's weight is at least $n-k+2$. By the Singleton bound the weight o f the code is $n-k+2$, and the code is MDS. □

**Problem 2.**
Proof: The formula in Theorem 8.5 of Wicker (p. 189) is:

$$A_w = \binom{n}{w}(q-1)\sum_{i=0}^{w-d_{\min}}(-1)^i\binom{w-1}{i}q^{w-i-d_{\min}}.$$

Since for MDS co des $d_{\min}=n-k+1$, $w-d_{\min}=n-t-d_{\min}=k-t-1$ and the above formula becomes:

$$\begin{aligned}
A_w &= \binom{n}{w}(q-1)\sum_{j=0}^{k-t-1}(-1)^j\binom{w-1}{j}q^{k-t-1-j} \\
&= \binom{n}{w}\left[\sum_{j=0}^{k-t-1}(-1)^j\binom{w-1}{j}q^{k-t-j} - \sum_{j=0}^{k-t-1}(-1)^j\binom{w-1}{j}q^{k-t-1-j}\right] \\
&= \binom{n}{w}\left[\sum_{j=0}^{k-t-1}(-1)^j\binom{w-1}{j}(q^{k-t-j}-1) - \sum_{j=0}^{k-t-1}(-1)^j\binom{w-1}{j}(q^{k-t-1-j}-1)\right] \\
&= \binom{n}{w}\left[\sum_{j=0}^{k-t-1}(-1)^j\binom{w-1}{j}(q^{k-t-j}-1) + \sum_{j=1}^{k-t}(-1)^j\binom{w-1}{j-1}(q^{k-t-j}-1)\right]
\end{aligned}$$

$$= \binom{n}{w}\left[(q^{k-t}-1)+\sum_{j=1}^{k-t-1}(-1)^j\left[\binom{w-1}{j}+\binom{w-1}{j-1}\right](q^{k-t-j}-1)\right]$$
$$= \binom{n}{w}\sum_{j=0}^{k-t-1}(-1)^j\binom{w}{j}(q^{k-t-j}-1),$$

and we get the formula derived in class by Prof. McEliece. □

**Problem 3.**

Solution: This problem doesn't have a fixed form of answer. And I give full score to any answer that makes sense.

Generally speaking, the fact that the procedure **Euclid** returns $\sigma(x)=1$ here means there are too many errors, and a robust algorithm should realize that now or later. If the algorithm is 'poor'—that is, it doesn't check if the received codewords are correctable—then it will use the following recursive formula

$$S_{j \bmod n} = -\sum_{i=1}^{d}\sigma_i S_{j-i}$$

to compute $S_{j \bmod n}$ for $j=r+1$ to n. Here *d* is the degree of $\sigma(x)$, so *d=0* and the values of $S_{j \bmod n}$ $(j=r+1,\cdots,n)$ will not be computed at all. That will lead to decoding error.

**Problem 4.**

(a) When $e_0=16$ and $e_1=1$, the decoder will return the codeword that contains the 15 un-erased received symbols as its corresponding symbols, which is different from the correct codeword. So the probability of decoder error is 1.

(b) When $e_0=15$ and $e_1=1$, if a decoder error occurs then $e_1^{'} \le (r-e_0)/2 \Rightarrow e_1^{'}=0$. Therefore the returned codeword has $n-e_0-e_1=15$ components in common with the correct codeword, which means the returned codeword is the same as the correct codeword and there is no decoder error, and that is a contradiction. Therefore, the probability of decoder error is 0.

(c) The positions of the erasures and errors don't affect our analysis below. So WLOG we suppose in the received codeword $(C_0,C_1,\cdots,C_{30})$, the first 14 components—$C_0,C_1,\cdots,C_{13}$—are erased, the two errors are in $C_{14}$ and $C_{15}$, and the last 15 components are correct.

If a decoder error occurs, then $e_1^{'} \le (r-e_0)/2 \Rightarrow e_1^{'} \le 1$. If $e_1^{'}=0$, or if $e_1^{'}=1$ and the position where the received codeword differs from the returned codeword is in $C_{14}$ or $C_{15}$, then again the returned codeword will have $n-e_0-e_1=15$ components in common with the correct codeword, which indicates there is no decoder error. So $e_1^{'}=1$ if a decoder error occurs, and the position where the received codeword differs from the returned codeword must be among $C_{16},C_{17},\cdots,C_{30}$.

Say the received codeword differs from the returned codeword in position $C_i$ $(16 \le i \le 30)$. There are 15 choices for $i$. Fix $i$, then no matter what the error in $C_{14}$ is, $C_{14}$ and the 14 components among $C_{16}, C_{17}, \cdots, C_{30}$ except $C_i$ determines the returned codeword—and thus determines the value of $C_{15}$. However, if there doesn't have to be a decoder error, then $C_{15}$ can take on $q-1$ values because the error in it is between 1 and $q-1$. So the probability of decoder error is

$$\frac{15}{q-1} = \frac{15}{31} \approx 0.484.$$