Solutions to Homework Assignment 2

Problem 1.
A code that can detect 3 errors has $d_{min} = 3 + 1 = 4$. This is the same as a single error correcting, double error detecting code. Such a code was designed in Problem 5 of Homework Assignment 1. For $k = 32$, the minimum redundancy needed turns out to be 7. For a proof of this fact, as well as an example of a parity check matrix that does the job, look in the Solutions to Homework Assignment 1.

Problem 2(a).
Length of the code = Number of columns in$\mathbf{H}$ = 7.
Redundancy = Number of rows = 4.
Dimension = 7-4= 3.
Since all the columns in the parity check matrix are non-zero and distinct, d_{min} is at least 3. Also, rows 4, 5 and 7 sum to $\mathbf{0}$, therefore (0,0,0,1,1,0,1) is a codeword and the code has minimum distance 3.

Problem 2(b).
Length of the code = 7, Dimension = 3, exactly as in part (a). Now, check that no subset of 3 columns or less sums to $\mathbf{0}$, and hence the code has d_{min} at least 4. Also, columns 1, 2, 3 and 5 sum to $\mathbf{0}$, showing that the code has minimum distance 4.

Problem 3(a).

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} p(y_i|x_i)$$

$$= \begin{cases} 0 & \text{if } x_i \neq y_i \text{ in any of the nonerased positions} \\ p^e(1-p)^{n-e} & \text{otherwise} \end{cases}$$

where e is the number of erasures.

From this, we see that $p(\mathbf{y}|\mathbf{x})$ is the same for all $\mathbf{x}$ which match with $\mathbf{y}$ in the nonerased positions, and is zero otherwise.

Therefore, the ML detector picks a codeword randomly from among the ones that match with $\mathbf{y}$ in the non-erased positions.

Problem 3(b).

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \prod_{i=1}^{n} p(y_i|x_i) \\ &= \begin{cases} 0 & \text{if } x_i = 0, y_i = 1 \text{ for any } i \\ p^e(1-p)^{n-z-e} & \text{otherwise} \end{cases} \end{aligned}$$

where e is the number of errors (only of the type 1 changing to a 0, since the other type were ruled out via the first case) and z is the number of zeros in $\mathbf{x}$.

Now, it is easy form this to see that $p(\mathbf{y}|\mathbf{x})$ is maximised by any $\mathbf{x}$ that has no disagreements with $\mathbf{y}$ in the places in which $\mathbf{x}$ has a 0, and the minimum number of disagreements possible in the other positions (because $p < 1/2$). If there are more than one such $\mathbf{x}$'s, then $p(\mathbf{y}|\mathbf{x})$ is the same for all of them.

Therefore, the ML decoder picks a codeword that has no disagreements with the received vector in the places that the codeword has 0's, and among all such codewords, it picks one with the minimim possible number of disagreements with the received vector.

Problem 4(a). Using the generator matrix, we can generate all possible codewords. To do this, take all length 4 binary vectors and multiply each of them by the generator matrix. On doing this, we see that the code has, in addition to the all zeros codeword, 14 codewords of weight 4 and one of weight 8. The minimum distance of a linear code is also the minimum weight among all nonzero codewords, and therefore this code has minimum distance 4.

Problem 4(b). Since the decoder only detects errors, the decoding algorithm consists of multiplying the received vector by the parity check matrix and declaring an error if the resulting syndrome is not the all-zeros vector. Thus, the algorithm will make a mistake only if the received vector is a codeword different from the transmitted one (since only codewords have syndrome zero). Equivalently, the algorithm makes an error if and only if the error pattern is a nonzero codeword.

The probability of this event can be computed using the weight distribution of the code computed in part (a), and is given by

$$P_{err} = 14p^4(1-p)^4 + p^8.$$