TEXTBOOK PROBLEMS

pg. 199: # 4, 6, 8, 12.

pg. 312/313: # 4, 16.

OTHER PROBLEMS OF AN EXPLORATORY NATURE

Binary Arithmetic.

Consider the set \{0, 1\} which we will call \(Z_2\). This is the set of binary numbers with the following addition and multiplication rules:

\[
\begin{array}{c|cc}
+ & 0 & 1 \\
\hline
0 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\quad \begin{array}{c|cc}
\cdot & 0 & 1 \\
\hline
0 & 0 & 0 \\
1 & 0 & 1 \\
\end{array}
\]

A binary vector space of dimension \(n\), denoted \(V_n\) is a collection of ordered \(n\)-tuples whose entries are either 0 or 1. For example,

\[
V_2 = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}
\]

\[
V_3 = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \ldots, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}
\]
Note that $V_n$ is a finite set and has $2^n$ elements. Each of these is a vector space with the set of scalars being $\mathbb{Z}_2$. Here are some examples of vector addition in $V_3$.

$$\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Note that addition follows the rules of addition in $\mathbb{Z}_2$ and the only scalars available to you are 0 and 1.

1. Write down a basis for the vector space $V_n$ described above.

A *binary code of length n* is simply a vector in $V_n$. For example, $(1, 0, 1)$ is a code of length 3; $(0, 1, 0, 1)$ is a binary code of length 4 and so on. In the real world, all telecommunication transmissions are nothing but transmissions of binary codes. Every word of email or text message you send is converted to a binary code of appropriate length and transmitted via cable or phone lines. Ergo, there is a possibility of error in the transmission. Here is an example:

You are sitting in a watch tower in Boston (and you a friend of Paul Revere). You are asked to transmit the binary code $(0, 1)$ if the British are coming by land, $(1, 1)$ if by sea, $(1, 0)$ if by air (the British have enlisted the help of Voldemort, obviously), and $(0, 0)$ if they don’t show up at all. Here are a couple of possibilities:

1. You transmit $(0, 1)$ and there is a transmission error in the cable line which results in a single change in the message; General Washington receives $(0, 0)$ which is obviously disastrous.

2. You transmit $(1, 1)$, and again due to a single transmission error, the message that is actually received is $(0, 1)$ and all the preparations are for a ground assault.

How do we fix this? Well, instead of a code of length 2, we can send a code of length 3, where we append a 0 or 1 so that the resulting binary code has an even number of 1’s. So our amended transmissions would look like:

$(0, 1, 1) \Rightarrow \text{land}, \quad (1, 1, 0) \Rightarrow \text{sea}, \quad (1, 0, 1) \Rightarrow \text{air}, \quad (0, 0, 0) \Rightarrow \text{not coming}$

2. Show that if there is a single transmission error, then the receiver will be able to detect it.
3. If the transmission received is $(1, 1, 1)$, then the receiver knows a single error has been made; can the receiver correct this error, i.e., can the receiver determine the original message?

4. So appending a "check" digit allows us to detect all single digit errors. Show that sending the whole message doubled as a vector in $V_4$ is an error correcting code, i.e., if we were to send $(0, 0, 0, 0), (0, 1, 0, 1), \ldots$ instead of $(0, 0), (0, 1), \ldots$ (repeating the message), then how that this will detect and correct all single digit transmission errors.

**Remark:** Note that sending a message of twice the length is indeed an error correcting code, but it is horribly inefficient. The subject of coding theory explores ways in which one can send the smallest possible message and still detect and correct as many errors as possible.