1. **Problem 5.8. Huffman coding.** Consider the random variable:

\[ X = \left( \begin{array}{cccccc}
    x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\
    0.49 & 0.26 & 0.12 & 0.04 & 0.04 & 0.03 & 0.02
  \end{array} \right) \]

**Solution:**

(a) Find a binary Huffman code for X.

![Figure 1: Binary Huffman code, prob. 5.8](image)

(b) Find the expected code length for this encoding.

\[
L(C) = 0.49(1) + 0.26(2) + 0.12(3) + 0.04(5) + 0.04(5) + 0.03(5) + 0.02(5)
\]

\[= 1.96\]

\[\approx 2\text{ bits}\]

(c) Find a ternary Huffman code for X.

![Figure 2: Ternary Huffman code, prob. 5.8](image)
2. **Problem 5.12. Shannon codes and Huffman codes.** Consider a random variable X that takes on four values with probabilities \( \left( \frac{1}{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{12} \right) \).

(a) **Construct a Huffman code for this random variable.**

<table>
<thead>
<tr>
<th>CODES</th>
<th>1</th>
<th>x1</th>
<th>1/3 ——— 1/3 ——— 2/3 ——— 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>x2</td>
<td>1/3 ——— 1/3 ——— 1/3 ——— 1/3</td>
<td></td>
</tr>
<tr>
<td>010</td>
<td>x3</td>
<td>1/4 ——— 1/3 ——— 1/3 ——— 1/3</td>
<td></td>
</tr>
<tr>
<td>011</td>
<td>x4</td>
<td>1/12 ——— 1/3 ——— 1/3 ——— 1/3</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: Binary Huffman code (1), prob. 5.12a

<table>
<thead>
<tr>
<th>CODES</th>
<th>00</th>
<th>x1</th>
<th>1/3 ——— 1/3 ——— 2/3 ——— 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>x2</td>
<td>1/3 ——— 1/3 ——— 1/3 ——— 1/3</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>x3</td>
<td>1/4 ——— 1/3 ——— 1/3 ——— 1/3</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>x4</td>
<td>1/12 ——— 1/3 ——— 1/3 ——— 1/3</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4: Binary Huffman code (2), prob. 5.12a

(b) **Show that there exist two different sets of optimal lengths for the codewords; namely, show that codeword length assignments \((1, 2, 3, 3)\) and \((2, 2, 2, 2)\) are both optimal.**

For lengths \((1, 2, 3, 3)\):

\[
L(C_1) = 1(1/3) + 2(1/3) + 3(1/4) + 3(1/12) \\
= 2
\]

Kraft Inequality test:

\[
2^{-1} + 2^{-2} + 2^{-3} + 2^{-3} \leq 1 \\
1 = 1
\]

For lengths \((2, 2, 2, 2)\):

\[
L(C_2) = 2(1/3) + 2(1/3) + 2(1/4) + 2(1/12) \\
= 2
\]

Kraft Inequality test:

\[
2^{-2} + 2^{-2} + 2^{-2} + 2^{-2} \leq 1 \\
1 = 1
\]
The two codes have the same expected length of 2 bits, and both satisfy Kraft inequality. So, the two codes are optimal.

(c) Conclude that there are optimal codes with codeword lengths for some symbols that exceed the Shannon code length $\lceil \log_2 \frac{1}{p(x)} \rceil$

First we need to compute the Shannon code lengths for the distribution:

<table>
<thead>
<tr>
<th>$p_i$</th>
<th>$\log_2 \frac{1}{p(x)}$</th>
<th>$\lceil \log_2 \frac{1}{p(x)} \rceil$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/3</td>
<td>1.58</td>
<td>1</td>
</tr>
<tr>
<td>1/3</td>
<td>1.58</td>
<td>2</td>
</tr>
<tr>
<td>1/4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1/12</td>
<td>3.58</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

Table 1: Computation of Shannon code lengths, problem 5.12

We can compare the lengths for each $x_i$:

<table>
<thead>
<tr>
<th>$x_i$</th>
<th>$p(x_i)$</th>
<th>Opt.Cod.1</th>
<th>Opt.Cod.2</th>
<th>Shanon</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>1/3</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1/3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$x_3$</td>
<td>1/4</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$x_4$</td>
<td>1/12</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 2: Lengths for Cod1, Cod2 and Shanon, problem 5.12

From the previous table, we can see that the length for $x_3$ is 3 for the optimal Huffman code 1, and it’s length 2 for the Shannon code. We know that the expected length of the Huffman code is minimum and less than the Shannon code. As shown in this problem, there might be a symbol in the Huffman code with length higher than the one in Shannon code.

3. Problem 5.17. Data compression. Find an optimal set of binary codeword lengths $l_1, l_2, ...$ (minimizing $\sum p_i l_i$) for an instantaneous code for each of the following probability mass functions:

(a) $p = (10/41, 9/41, 8/41, 7/41, 7/41)$

We know that the code with the minimum expected length is the Huffman code, then:

<table>
<thead>
<tr>
<th>CODES</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>10/41</td>
<td>14/41</td>
<td>17/41</td>
<td>24/41</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>9/41</td>
<td>10/41</td>
<td>14/41</td>
<td>17/41</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>8/41</td>
<td>9/41</td>
<td>10/41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000</td>
<td>7/41</td>
<td>8/41</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>001</td>
<td>7/41</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5: Binary Huffman Code, prob. 5.17
Now we can use the Kraft inequality to check whether this code (with lengths: 2, 2, 2, 3, 3) satisfy it:

\[ 2^{-2} + 2^{-2} + 2^{-2} + 2^{-3} + 2^{-3} \leq 1 \]
\[ 1 = 1 \]

This set of code lengths satisfy the Kraft inequality and since it has the minimum expected length (Huffman code), this code is optimal.

**(b)** \( p = (\frac{9}{10}, (\frac{9}{10})(\frac{1}{10})^1, (\frac{9}{10})(\frac{1}{10})^2, (\frac{9}{10})(\frac{1}{10})^3, \ldots) \)

Define \( p = (p_1, p_2, p_3, \ldots p_n) \).

From the given data we have: \( p_1 > p_2 > p_3 > \ldots > p_{n-2} > p_{n-1} > p_n \).

Since: \( \frac{9}{10} > (\frac{9}{10})(\frac{1}{10})^1 > (\frac{9}{10})(\frac{1}{10})^2 > (\frac{9}{10})(\frac{1}{10})^3 > \ldots > (\frac{9}{10})(\frac{1}{10})^{n-2} > (\frac{9}{10})(\frac{1}{10})^{n-1} > (\frac{9}{10})(\frac{1}{10})^n \)

For constructing the Huffman code, we need to group the two lowest probabilities \( (p_{n-1} + p_n) \) for the first stage of the coding process. Then, we need to find its the position in the second column. Since we know the exact number corresponding to each probability we do the following analysis:

\[
p_{n-1} + p_n = (\frac{9}{10})(\frac{1}{10})^{n-1} + (\frac{9}{10})(\frac{1}{10})^n \\
= (\frac{9}{10})(\frac{1}{10})^{n-1} + (\frac{1}{10})^n \\
= (\frac{9}{10})(\frac{1}{10})^n(11) \\
= (\frac{9}{10})(\frac{1}{10})^n(100) \]

Now we analyze the closest higher probability to this group:

\[
p_{n-2} = (\frac{9}{10})(\frac{1}{10})^{n-2} \\
= (\frac{9}{10})(\frac{1}{10})^n(100) \]

From this analysis we obtain that: \( p_{n-1} + p_n < p_{n-2} \).

This analysis is valid for all \( n \):

\[
p_{n-1} + p_n < p_{n-2} \\
(\frac{9}{10})(\frac{1}{10})^n(11) < (\frac{9}{10})(\frac{1}{10})^n(100) \\
11 < 100 \]
Hence we can construct a Binary Huffman code as follows:

Figure 6: Binary Huffman code for problem 5.17b

Then, the codewords lengths \((l_i)\) are given by: \((i, i - 1 + 1, i - 1 + 1, \ldots, n - 1, n - 1)\)

4. **Problem 5.30.** *Relative entropy is cost of miscoding.* Let the random variable \(X\) have five possible outcomes \(\{1, 2, 3, 4, 5\}\). Consider two distributions \(p(x)\) and \(q(x)\) on this random variable.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>(p(x))</th>
<th>(q(x))</th>
<th>(C_1(x))</th>
<th>(C_2(x))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1/2)</td>
<td>(1/2)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>(1/4)</td>
<td>(1/8)</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>(1/8)</td>
<td>(1/8)</td>
<td>110</td>
<td>101</td>
</tr>
<tr>
<td>4</td>
<td>(1/16)</td>
<td>(1/8)</td>
<td>1110</td>
<td>110</td>
</tr>
<tr>
<td>5</td>
<td>(1/16)</td>
<td>(1/8)</td>
<td>1111</td>
<td>111</td>
</tr>
</tbody>
</table>

Table 3: Problem 5.30

**(a) Calculate** \(H(p)\), \(H(q)\), \(D(p||q)\), and \(D(q||p)\).

\[
H(p) = \frac{1}{2} \log_2 2 + \frac{1}{4} \log_2 4 + \frac{1}{8} \log_2 8 + \frac{1}{16} \log_2 16 + \frac{1}{16} \log_2 16 \tag{22}
\]
\[
= \frac{1}{2} + \frac{1}{2} + \frac{3}{8} + \frac{1}{4} + \frac{1}{4} \tag{23}
\]
\[
= \frac{15}{8} \tag{24}
\]

\[
H(q) = \frac{1}{2} \log_2 2 + \frac{1}{8} \log_2 8 + \frac{1}{8} \log_2 8 + \frac{1}{8} \log_2 8 + \frac{1}{8} \log_2 8 \tag{25}
\]
\[
= \frac{1}{2} + 4 \left(\frac{3}{8}\right) \tag{26}
\]
\[
= 2 \tag{27}
\]
\[ D(p||q) = \sum p(x) \log_2 \frac{p(x)}{q(x)} \]  
\[ = \frac{1}{2} \log_2 \frac{1/2}{1/2} + \frac{1}{4} \log_2 \frac{1/4}{1/8} + \frac{1}{8} \log_2 \frac{1/8}{1/16} + \frac{1}{16} \log_2 \frac{1/16}{1/16} \]  
\[ = \frac{1}{8} \]  
\[ D(q||p) = \sum q(x) \log_2 \frac{q(x)}{p(x)} \]  
\[ = \frac{1}{2} \log_2 \frac{1/2}{1/2} + \frac{1}{4} \log_2 \frac{1/4}{1/4} + \frac{1}{8} \log_2 \frac{1/8}{1/8} + \frac{1}{16} \log_2 \frac{1/16}{1/16} \]  
\[ = \frac{1}{8} \]  

(b) The last two columns represent codes for the random variable. Verify that the average length of \( C_1 \) under \( p \) is equal to the entropy \( H(p) \). Thus, \( C_1 \) is optimal for \( p \). Verify that \( C_2 \) is optimal for \( q \).

First we check for \( C_1(x) \):

\[ L(C_1) = \sum_i (p(x_i)) l_i \]  
\[ = \frac{1}{2}(1) + \frac{1}{4}(2) + \frac{1}{8}(3) + \frac{1}{16}(4) + \frac{1}{16}(4) \]  
\[ = \frac{1}{2} + \frac{1}{2} + \frac{3}{8} + \frac{1}{4} + \frac{1}{4} \]  
\[ = \frac{15}{8} \]  

Hence, we have that \( L(C_1) = H(p) = \frac{15}{8} \).

Now we check for \( C_2(x) \):

\[ L(C_2) = \sum_i (q(x_i)) l_i \]  
\[ = \frac{1}{2}(1) + \frac{1}{8}(2) + \frac{1}{8}(3) + \frac{1}{8}(3) + \frac{1}{8}(3) \]  
\[ = \frac{1}{2} + 4\left(\frac{3}{8}\right) \]  
\[ = 2 \]  

Hence, we have that \( L(C_2) = H(q) = 2 \).
(c) Now assume that we use code $C_2$ when the distribution is $p$. What is the average length of the codewords. By how much does it exceed the entropy $p$?

$$L(C_2)_{p(x)} = \sum_i (p(x_i))(l_i)$$  \hspace{1cm} (42)

$$= \frac{1}{2}(1) + \frac{1}{4}(3) + \frac{1}{8}(3) + \frac{1}{16}(3) + \frac{1}{16}(3)$$  \hspace{1cm} (43)

$$= \frac{1}{2} + \frac{3}{4} + \frac{3}{8} + \frac{3}{16} + \frac{3}{16}$$  \hspace{1cm} (44)

$$= 2$$  \hspace{1cm} (45)

To see how much it does exceed the entropy $H(p)$ we compute:

$$L(C_2)_{p(x)} - H(p) = 2 - \frac{15}{8}$$  \hspace{1cm} (46)

$$= \frac{1}{8}$$  \hspace{1cm} (47)

(d) What is the loss if we use code $C_1$ when the distribution is $q$?

$$L(C_1)_{q(x)} = \sum_i (q(x_i))(l_i)$$  \hspace{1cm} (48)

$$= \frac{1}{2}(1) + \frac{1}{8}(2) + \frac{1}{8}(3) + \frac{1}{8}(4) + \frac{1}{8}(4)$$  \hspace{1cm} (49)

$$= \frac{1}{2} + \frac{1}{4} + \frac{3}{8} + \frac{1}{2} + \frac{1}{2}$$  \hspace{1cm} (50)

$$= 17/18$$  \hspace{1cm} (51)

To see how much the loss is, we compute:

$$L(C_1)_{q(x)} - H(q) = 17/8 - 2$$  \hspace{1cm} (52)

$$= \frac{1}{8}$$  \hspace{1cm} (53)

For (d) and (c) we can see that the loss is given by $D(p||q)$ and $D(q||p)$ correspondingly. (The wrong distribution!).

5. Problem 5.32. Bad wine. One is given six bottles of wine. It is known that precisely one bottle has gone bad (tastes terrible). From inspection of the bottles it is determined that the probability $p_i$ that the $i$th bottle is bad is given by $(p_1, p_2, ..., p_6) = (\frac{8}{23}, \frac{6}{23}, \frac{4}{23}, \frac{2}{23}, \frac{2}{23}, \frac{1}{23})$. Tasting will determine the bad wine. Suppose that you taste the wines one at a time. Choose the order of tasting to minimize the expected number of tastings required to determine the bad bottle. Remember, if the first five wines pass the test, you dont have to taste the last.
(a) What is the expected number of tastings required?

Since we know the probability of the ith bottle to be the one containing the bad wine, we should start tasting the bottles in decreasing probability order, starting from the one with higher probability and so on.

So, we have the probability distribution: \((p_1, p_2, ..., p_6) = \left(\frac{8}{23}, \frac{6}{23}, \frac{4}{23}, \frac{2}{23}, \frac{2}{23}, \frac{1}{23}\right)\)

\[
Number_{Tastes} = (1) \frac{8}{23} + (2) \frac{6}{23} + (3) \frac{4}{23} + (4) \frac{2}{23} + (5) \frac{2}{23} + (5) \frac{1}{23}
\]

\[
= \frac{55}{23}
\]

(b) Which bottle should be tasted first?

The bottle that should be tasted first is the one with the higher probability of containing the bad wine \((p_1 = \frac{8}{23})\).

Now you get smart. For the first sample, you mix some of the wines in a fresh glass and sample the mixture. You proceed, mixing and tasting, stopping when the bad bottle has been determined.

(a) What is the minimum expected number of tastings required to determine the bad wine?

Since we know that the Huffman code provides the minimum expected length, we can use it for solving this problem. Then, we compute the expected length as follows:

<table>
<thead>
<tr>
<th>CODES</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>8/23</td>
<td>8/23</td>
<td>8/23</td>
<td>9/23</td>
<td>14/23</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>01</td>
<td>6/23</td>
<td>6/23</td>
<td>6/23</td>
<td>8/23</td>
<td>9/23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>2/23</td>
<td>3/23</td>
<td>4/23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>2/23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>1/23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 7: Huffman code applied to problem 5.32

\[
L(C) = \sum_i p(x_i)(l_i)
\]

\[
= (2) \frac{8}{23} + (2) \frac{6}{23} + (2) \frac{4}{23} + (3) \frac{2}{23} + (4) \frac{2}{23} + (4) \frac{1}{23}
\]

\[
= \frac{54}{23}
\]

\[
\approx 3
\]
The minimum expected number of tastes is 3.

(b) What mixture should be tasted first?

Seeing the diagram, one can use the paths of the Huffman code to make the mixtures by grouping the probabilities. Thus, the bottles with probabilities 8/23 and 6/23 form the first mixture (given by the path leading to the 14/23 probability, first group). This mixture should be tasted first since it has the higher probability. If the bad wine is not in this mixture we know the bad wine must be in one of the remaining bottles (the path with the one leading to the probability 9/23). The second mixture is formed by the bottles with probabilities 4/23 and 2/23. After tasting this mixture and haven’t found the bad wine only one more taste is needed to find the bad wine.