Solutions to Practice Midterm
(Total: 160 points)

There are 3 problems, each problem with multiple parts, each part worth 10 points. Your answer should be as clear and readable as possible.

1. A dyadic source (70 pts). Consider a source that produces an i.i.d. sequence of symbols from the alphabet \{A, B, C, D, E, F\} with probabilities 1/2, 1/8, 1/8, 1/8, 1/16, 1/16.

(a) Find the (binary) entropy of the source.
(b) Find the Huffman code that takes 1 source symbol at a time.
(c) Find the average codeword length of the code constructed in part (b).
(d) Can you improve the compression rate (measured by the average codeword length per source symbol) by taking 2 source symbols at a time? Justify your answer.
(e) Find the Shannon–Fano code that takes 1 source symbol at a time.
(f) Find the average codeword length of the code constructed in part (e).
(g) (Difficult.) Is the Shannon–Fano code always optimal for sources with dyadic probabilities (that is, the probabilities of the form $2^{-k}$)? Prove it or provide a counterexample.

Solution:

(a) By the definition of binary entropy, we have

\[
H = \frac{1}{2} \log_2 2 + 3 \times \frac{1}{8} \log_2 8 + 2 \times \frac{1}{16} \log_2 16 = 2.125.
\]

(b) The Huffman code is given as follows.

<table>
<thead>
<tr>
<th>Code</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>A 1/2</td>
</tr>
<tr>
<td>110</td>
<td>B 1/8</td>
</tr>
<tr>
<td>111</td>
<td>C 1/8</td>
</tr>
<tr>
<td>100</td>
<td>D 1/8</td>
</tr>
<tr>
<td>1010</td>
<td>E 1/16</td>
</tr>
<tr>
<td>1011</td>
<td>F 1/16</td>
</tr>
</tbody>
</table>

1
(c) The codeword length is
\[
L_H = \frac{1}{2} + 3 \times \left( \frac{1}{8} + \frac{1}{8} + \frac{1}{8} \right) + 4 \times \left( \frac{1}{16} + \frac{1}{16} \right) = 2.125.
\]

(d) We cannot improve the compression rate further, because the codeword length is already equal to the entropy.

(e) The Shannon–Fano code is given as follows.

<table>
<thead>
<tr>
<th>Code</th>
<th>0</th>
<th>1/2</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>B</td>
<td>1/8</td>
<td>0</td>
</tr>
<tr>
<td>101</td>
<td>C</td>
<td>1/8</td>
<td>1</td>
</tr>
<tr>
<td>110</td>
<td>D</td>
<td>1/8</td>
<td>1</td>
</tr>
<tr>
<td>1110</td>
<td>E</td>
<td>1/16</td>
<td>0</td>
</tr>
<tr>
<td>1111</td>
<td>F</td>
<td>1/16</td>
<td>1</td>
</tr>
</tbody>
</table>

(f) The average codeword length is
\[
\bar{L}_{S-F} = \frac{1}{2} + 3 \times \left( \frac{1}{8} + \frac{1}{8} + \frac{1}{8} \right) + 4 \times \left( \frac{1}{16} + \frac{1}{16} \right) = 2.125.
\]

(g) Yes, the Shannon–Fano code is always optimal for sources with dyadic probabilities. Let \(2^{-k_1} \geq 2^{-k_2} \geq \cdots \geq 2^{-k_n}\) be the dyadic probabilities. We first note that the optimal code should assign length \(k_i\) to the symbol with probability \(2^{-k_i}\). To see the optimality of the Shannon–Fano code in this case, we establish the following lemma.

**Lemma 1.** There exists \(1 \leq n \leq m\) such that \(\sum_{i=1}^{n} 2^{-k_i} = 1/2\).

**Proof.** Suppose the lemma is not true. Then we must have
\[
\sum_{i=1}^{n} 2^{-k_i} < 2^{-1} \tag{1}
\]
and
\[
2^{-k_{n+1}} + \sum_{i=1}^{n} 2^{-k_i} > 2^{-1} \tag{2}
\]
for some \(1 \leq n \leq m\). This implies
\[
2^{-k_{n+1}} \overset{(a)}{>} 2^{-1} - \sum_{i=1}^{n} 2^{-k_i} \overset{(b)}{=} \frac{2^{k_n} - \sum_{i=1}^{n} 2^{k_n-k_i}}{2^{k_n}} \overset{(c)}{=} \frac{1}{2^{k_n}}.
\]
where \((a)\) follows by moving the term \(\sum_{i=1}^{n} 2^{-k_i}\) to the right hand side of equation (2), \((b)\) follows by multiplying \(2^{k_n}\) on both the numerator and denominator, and \((c)\) follows since equation (1) forces the numerator of \((b)\) to be a positive integer. However, this contradicts the assumption that \(2^{-k_{n+1}} \leq 2^{-k_n}\). □

As a consequence of the lemma, we can split the probabilities at exactly half in the first step of Shannon–Fano coding. Now consider the two parts \(\{2^{-k_1}, \ldots, 2^{-k_n}\}\) and \(\{2^{-k_n+1}, \ldots, 2^{-k_m}\}\). Dividing by \(1/2\), they become two sets of dyadic probabilities \(\{2^{-(k_1-1)}, \ldots, 2^{-(k_n-1)}\}\) and \(\{2^{-(k_{n+1}-1)}, \ldots, 2^{-(k_m-1)}\}\). Applying the lemma again, each set can be split at exactly half. We repeat this procedure until the largest probability in a set becomes 1. Therefore, we are always able to split the probabilities at exactly half at each step of Shannon–Fano coding. One can check that the number of splits it takes for probability \(2^{-k_i}\) to reach 1 is \(k_i\), which implies that a length \(k_i\) codeword is assigned to this symbol. This is exactly the optimal codeword length.
2. Hierarchical Huffman coding (40 pts).

Consider a source that produces an i.i.d. sequence of symbols from the alphabet \{A, B, C, D, E, F, G\} with probabilities

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\frac{1}{28} & \frac{2}{28} & \frac{3}{28} & \frac{4}{28} & \frac{5}{28} & \frac{6}{28} & \frac{7}{28} \\
\end{array}
\]

A clever engineer would like to construct the binary Huffman code, but would like to find a shortcut.

(a) Find the quaternary Huffman code with code alphabet \{\alpha, \beta, \gamma, \delta\} that takes 1 source symbol at a time.

(b) From the quaternary Huffman code found in part (a), construct a binary code by mapping

\[
\alpha \rightarrow 00, \quad \beta \rightarrow 01, \quad \gamma \rightarrow 10, \quad \delta \rightarrow 11.
\]

(c) Is the code in part (b) instantaneous? Justify your answer.

(d) Is the code in part (b) optimal (that is, does it achieve the shortest average codeword length among all binary instantaneous codes)? Prove it or provide a counterexample.

Solution:

(a) The quaternary Huffman code is given as follows.

<table>
<thead>
<tr>
<th>Code</th>
<th>quaternary</th>
<th>binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>\alpha\delta</td>
<td>0011</td>
</tr>
<tr>
<td>B</td>
<td>\alpha\gamma</td>
<td>0010</td>
</tr>
<tr>
<td>C</td>
<td>\alpha\beta</td>
<td>0001</td>
</tr>
<tr>
<td>D</td>
<td>\alpha\alpha</td>
<td>0000</td>
</tr>
<tr>
<td>E</td>
<td>\delta</td>
<td>11</td>
</tr>
<tr>
<td>F</td>
<td>\gamma</td>
<td>10</td>
</tr>
<tr>
<td>G</td>
<td>\beta</td>
<td>01</td>
</tr>
</tbody>
</table>

(b) The binary code constructed through the mapping is given as follows.
(c) Yes, the code in part (b) is instantaneous, because the codewords are the leaves of the following tree.

(d) No, the code is not optimal. Consider the following binary Huffman code.

```
<table>
<thead>
<tr>
<th>Code</th>
<th>01</th>
<th>00</th>
<th>000</th>
<th>001</th>
<th>110</th>
<th>1110</th>
<th>1111</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>G</td>
<td>F</td>
<td>E</td>
<td>D</td>
<td>C</td>
<td>B</td>
<td>A</td>
</tr>
</tbody>
</table>

Denote the average codeword lengths of the above binary Huffman code and the code in part (b) as \( L_B \) and \( L_{QB} \) respectively. We have

\[
L_B = 2 \times \left( \frac{7}{28} + \frac{6}{28} \right) + 3 \times \left( \frac{5}{28} + \frac{4}{28} + \frac{3}{28} \right) + 4 \times \left( \frac{2}{28} + \frac{1}{28} \right) = \frac{74}{28} = 2.643
\]

and

\[
L_{QB} = 2 \times \left( \frac{7}{28} + \frac{6}{28} + \frac{5}{28} \right) + 4 \times \left( \frac{4}{28} + \frac{3}{28} + \frac{2}{28} + \frac{1}{28} \right) = \frac{76}{28} = 2.714 > L_B.
\]
3. **Scalar quantization (50 pts).** Consider a source with the probability density function

\[
f(x) = \begin{cases} 
  1/7, & -5 \leq x \leq 1 \text{ or } 4 \leq x \leq 5, \\
  0, & \text{otherwise}. 
\end{cases}
\]

A clever engineer would like to construct the optimal 1-bit scalar quantizer by specifying two quantization regions \([b_0 = -5, b_1]\) and \((b_1, b_2 = 5]\), and two quantization points \(a_1 \leq b_1\) and \(a_2 > b_1\).

(a) Starting with \(b_1 = 0\), find the optimal \(a_1\) and \(a_2\) that minimize the mean squared error.

(b) Given \(a_1\) and \(a_2\) found in part (a), find the optimal \(b_1\).

(c) Given \(b_1\) found in part (b), find the optimal \(a_1\) and \(a_2\).

(d) What is the mean squared error of the quantizer constructed in part (c).

(e) Is the quantizer constructed in part (c) optimal? Prove it or provide a counterexample.

**Solution:**

(a) Applying the necessary condition for the optimal quantizer, we have

\[
a_1 = \frac{\int_{b_0}^{b_1} x f(x) dx}{\int_{b_0}^{b_1} f(x) dx} = \frac{\int_{-5}^{0} \frac{x}{7} dx}{\int_{-5}^{0} \frac{1}{7} dx} = -2.5
\]

and

\[
a_2 = \frac{\int_{b_1}^{b_2} x f(x) dx}{\int_{b_1}^{b_2} f(x) dx} = \frac{\int_{1}^{5} \frac{x}{7} dx + \int_{4}^{5} \frac{1}{7} dx}{\int_{1}^{5} \frac{1}{7} dx + \int_{4}^{5} \frac{1}{7} dx} = 2.5.
\]

(b) Given \(a_1\) and \(a_2\) in part (a), the optimal \(b_1\) is

\[
b_1 = \frac{a_1 + a_2}{2} = 0.
\]

(c) Given \(b_1 = 0\), the computation is the same as part (a). So \(a_1 = -2.5\) and \(a_2 = 2.5\).

(d) The mean squared error is

\[
\varepsilon^2 = \int_{b_0}^{b_1} (x - a_1)^2 f(x) dx + \int_{b_1}^{b_2} (x - a_2)^2 f(x) dx
\]

\[
= \int_{-5}^{0} \frac{1}{7} (x + 2.5)^2 dx + \int_{1}^{4} \frac{1}{7} (x - 2.5)^2 dx + \int_{4}^{5} \frac{1}{7} (x - 2.5)^2 dx
\]

\[
= 2.655.
\]
(e) No, the quantizer constructed in part (c) is not optimal. We observe that the pdf $f(x)$ has two nonzero parts. Let $a_1 = -2$ be the centroid of the left nonzero part $[-5, 1]$ and $a_2 = 4.5$ be the centroid of the right nonzero part $[4, 5]$. We repeat the procedure above. The resultant quantizer is given by

$$a_1 = -2, \ a_2 = 4.5, \ b_1 = 1.25.$$ 

The corresponding mean squared error is

$$\varepsilon^2 = \int_{-5}^{1} \frac{1}{7} (x + 2)^2 dx + \int_{4}^{5} \frac{1}{7} (x - 4.5)^2 dx = 2.583 < 2.655.$$