1. Consider

a. a continuous source with the following probability density function. Construct an optimal 2 bit/sample scalar quantizer for this source.

From notes:

\[ b_0 = 0, \quad b_4 = 2. \]

\[
\begin{align*}
    b_j &= \frac{a_j + a_{j+1}}{2} \\
    q_j &= \frac{\int_{b_{j-1}}^{b_j} x f_X(x) \, dx}{\int_{b_{j-1}}^{b_j} f_X(x) \, dx}
\end{align*}
\]

By symmetry, \( b_2 = 1 \).

So what are are

\[ b_3 = 2 - b_1. \]

left to compute are \( a_1, b_1, a_2 \).

\[ a_1 = \frac{\int_{0}^{b_1} x^2 \, dx}{\int_{0}^{b_1} x \, dx} = \frac{2}{3} b_1, \quad a_3 = \frac{\int_{b_1}^{1} x^2 \, dx}{\int_{b_1}^{1} x \, dx} = \frac{2}{3} \frac{1 - b_1^3}{1 - b_1^2} = \frac{2}{3} \frac{1 + b_1 + b_1^2}{b_1 + 1} \]

However

\[ b_1 = \frac{a_1 + a_2}{2} \Rightarrow b_1 = \frac{1}{3} b_1 + \frac{1}{3} \frac{1 + b_1 + b_1^2}{b_1 + 1} \Rightarrow 2b_1(b_1 + 1) = 1 + b_1 + b_1^2 \]

\[ \Rightarrow b_1^2 + b_1 - 1 = 0 \]

\[ b_1 = \frac{-1 \pm \sqrt{1 + 4}}{2} \Rightarrow b_1 = \frac{\sqrt{5} - 1}{2} \]

\[ a_1 = \frac{\sqrt{5} - 1}{3} \quad \text{and} \quad a_2 = \frac{\sqrt{5} - 1}{2} + \frac{\sqrt{5} - 1}{6} \]
b. a source with $m = 5$ symbols occurring with probabilities $\{p_1, p_2, \ldots, p_m\}$ whose first two most likely symbols have been encoded into codewords of lengths 1 and 2 bits each. What are the shortest codewords that can be assigned to the remaining symbols such that the resulting code is instantaneous. What is the minimum average length per symbol, $\bar{L}$, in terms of $\{p_1, p_2, \ldots, p_5\}$?

(23 points)

From Kraft's Inequality

$$-\frac{1}{2} - \frac{1}{2} - \frac{1}{2} - \frac{1}{2} - \frac{1}{2} \leq 1$$

\[\frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} \leq 1\]

\[\frac{1}{2} + \frac{1}{2} + \frac{1}{2} \leq \frac{1}{4}\]

Smallest integers satisfying this are $l_3 = 3$, $l_4 = l_5 = 4$.

So the average bits/symbol $\bar{L} = p_1 + 2p_2 + 3p_3 + 4(p_4 + p_5)$.

(assume $p_1 \geq p_2 \geq \ldots \geq p_m$; otherwise we order them).
Midterm

1. Consider a source which produces an i.i.d. sequence of symbols from the alphabet \{A, B, C, D, E, F, G\} with probabilities \{0.3, 0.25, 0.15, 0.1, 0.1, 0.05, 0.05\} respectively.

   a. Find binary Huffman code.

   \begin{align*}
   \text{Code} & \quad A & \quad B & \quad C & \quad D & \quad E & \quad F & \quad G \\
   \text{00} & \quad 0.3 & \quad 0.3 & \quad 0.3 & \quad 0.3 & \quad 0.25 & \quad 0.15 & \quad 0.15 \\
   \text{01} & \quad 0.25 & \quad 0.18 & \quad 0.25 & \quad 0.35 & \quad 0.15 & \quad 0.15 & \quad 0.15 \\
   \text{100} & \quad 0.15 & \quad 0.15 & \quad 0.25 & \quad 0.25 & \quad 0.15 & \quad 0.15 & \quad 0.15 \\
   \text{101} & \quad 0.1 & \quad 0.1 & \quad 0.1 & \quad 0.1 & \quad 0.1 & \quad 0.1 & \quad 0.1 \\
   \text{110} & \quad 0.05 & \quad 0.05 & \quad 0.05 & \quad 0.05 & \quad 0.05 & \quad 0.05 & \quad 0.05 \\
   \text{111} & \quad 0.05 & \quad 0.05 & \quad 0.05 & \quad 0.05 & \quad 0.05 & \quad 0.05 & \quad 0.05 \\
   \end{align*}

   b. Compute the average number of binary code symbols per source symbol, and the efficiency of this code.

   \[
   \bar{L} = (0.3 \times 0.25) \times 2 + (0.15 \times 0.1) \times 3 + (0.05 \times 0.05) \times 4 = 2.55.
   \]

   The entropy of the source is

   \[
   H = -0.3 \log 0.3 - 0.25 \log 0.25 - 0.15 \log 0.15 - 0.1 \log 0.1 - 0.1 \log 0.1 - 0.05 \log 0.05 - 0.05 \log 0.05
   \]

   \[
   = 2.528.
   \]

   So the efficiency of the code is \(\frac{H}{\bar{L}} = 99\%\).
2. Consider a source consisting of \( M = 32 \) symbols where symbol \( s_0 \) has a significantly larger probability than all other symbols. If \( s_0 \) is assigned a code word of length 1, what is the maximum number of code words of length 2? Assuming at least one codeword is of length 2 and source symbols \( s_0 \) and \( s_1 \) have the largest probabilities \( p_0 \) and \( p_1 \), what is the minimum average length of the code in terms of \( p_0 \) and \( p_1 \)?

Let the number of length-2 codewords be \( k \). By Kraft's inequality

\[
2^{-1} + k \cdot 2^{-2} \leq 1,
\]

Therefore \( k \leq 2 \). So the maximum is 2.

The minimum average length of the code is attained when only 3 symbols have positive probability. In this case, we assign length 1 to symbol \( s_0 \), length 2 to symbol \( s_1 \), and the third symbol. The corresponding average length of the code is

\[
\bar{L} = p_0 + 2p_1 + 2(1-p_0-p_1) = 2 - p_0.
\]
3. Consider a binary source $S$ with symbols $\{A, B\}$ with $\text{Prob}(B) = p$.
   
   a. Compute the source entropy.
   
   $$H(S) = -p \log_2 p - (1-p) \log_2 (1-p)$$
   
   b. Consider the following variable to fixed length coding scheme:

<table>
<thead>
<tr>
<th>Source Output</th>
<th>Code Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B$</td>
<td>000</td>
</tr>
<tr>
<td>$AB$</td>
<td>001</td>
</tr>
<tr>
<td>$AAB$</td>
<td>010</td>
</tr>
<tr>
<td>$AAAB$</td>
<td>011</td>
</tr>
<tr>
<td>$AAAAAB$</td>
<td>100</td>
</tr>
<tr>
<td>$AAAAAAB$</td>
<td>101</td>
</tr>
<tr>
<td>$AAAAAAAB$</td>
<td>110</td>
</tr>
<tr>
<td>$AAAAAAA$</td>
<td>111</td>
</tr>
</tbody>
</table>

   Calculate the average number of binary code digits per source digit.

   The average source symbol length is
   
   $$\bar{L} = p + 2(1-p)p + 3(1-p)^2p + 4(1-p)^3p + 5(1-p)^4p + 6(1-p)^5p$$
   
   $$+ 7(1-p)^6p + 7(1-p)^7$$
   
   $$= p + 2(1-p)p + 3(1-p)^2p + 4(1-p)^3p + 5(1-p)^4p + 6(1-p)^5p + 7(1-p)^6$$
   
   $$= p + 2(1-p)p + 3(1-p)^2p + 4(1-p)^3p + 5(1-p)^4p + 6(1-p)^5p + 7(1-p)^6$$
   
   $$= p + 2(1-p)p + 3(1-p)^2p + 4(1-p)^3p + 5(1-p)^4p + 6(1-p)^5p + 7(1-p)^6$$
   
   $$= p + 2(1-p)p + 3(1-p)^2p + 4(1-p)^3p + 5(1-p)^4p + 6(1-p)^5p + 7(1-p)^6$$
   
   $$= p + 2(1-p)p + 3(1-p)^2p + 4(1-p)^3p + 5(1-p)^4p + 6(1-p)^5p + 7(1-p)^6$$
   
   $$= 1 + (1-p) + (1-p)^2 + (1-p)^3 + (1-p)^4 + (1-p)^5 + (1-p)^6$$
   
   $$> \frac{1 - (1-p)^7}{1 - (1-p)}$$
   
   $$= \frac{1 - (1-p)^7}{p}$$
   
   The average number of binary code digits per source digit is $\frac{3}{\bar{L}} = \frac{3p}{1 - (1-p)^2}$
c. If \( p \) is close to 0, the source will emit long strings of \( A \)s, separated by \( B \)s. A typical output sequence of the source would be:

\[
\text{AABAABAGBBBAAAAABAB}\text{C}001000010110000101....
\]

Let us consider a new source \( S_2 \) consisting of alphabet \( \{s_0, s_1, \ldots, s_i, \ldots \} \) which emits the symbol \( s_i \) when the original source \( S \) emits a run of \( i \) \( A \)s followed by a \( B \). That is for the above binary output of the original source, the source \( S_2 \) emits the sequence:

\[
S_2s_4s_1s_8s_0s_8s_1...
\]

Find the entropy (base 2) of this new source \( S \). Do not leave your answer in terms of an infinite series but rather evaluate the infinite sum. The answer should be a function of \( p \).

By construction of the new source, we have

\[
P(s_i) = P(A^i B) = (1-p)^i p , \ i = 0, 1, 2, \ldots
\]

The entropy of the new source is

\[
H(S_2) = \sum_{i=0}^{\infty} -p(s_i) \log p(s_i)
\]

\[
= \sum_{i=0}^{\infty} - (1-p)^i p \log [(1-p)^i p]
\]

\[
= -p \log (1-p) \sum_{i=0}^{\infty} (1-p)^i - p \log p \sum_{i=0}^{\infty} (1-p)^i
\]

\[
\text{by hint}
\]

\[
= -p \log (1-p) \cdot \frac{(1-p)}{(1-(1-p))^2} - p \log p \cdot \frac{1}{1-(1-p)}
\]

\[
= \frac{1}{p} \left[-(1-p) \log (1-p) - p \log p \right]
\]

\[
= \frac{1}{p} H(S)
\]

**Hint:** \( \sum_{i=0}^{\infty} a^i = \frac{1}{1-a} \) for \( a < 1 \), and \( \sum_{i=0}^{\infty} i a^i = \frac{d}{da} \left( \sum_{i=0}^{\infty} a^i \right) = \frac{1}{(1-a)^2} \).
4. Consider a Lempel-Ziv compression algorithm with a window size of 16. Suppose the encoder has already encoded the following text:

**MY.BROTHER.TOM.ATE**

and is left to encode the remainder of the sentence which reads as

**.A.TOMATO**

Indicate the sequence of phrases that are encoded. Count the number of bits the remainder of the sentence is encoded into.

<table>
<thead>
<tr>
<th>Code</th>
<th># of bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 3, 2)</td>
<td>1+4+2</td>
</tr>
<tr>
<td>(1, 9, 4)</td>
<td>1+4+4</td>
</tr>
<tr>
<td>(1, 8, 2)</td>
<td>1+4+2</td>
</tr>
<tr>
<td>(1, 3, 1)</td>
<td>1+4+1</td>
</tr>
</tbody>
</table>

The number of bits the remainder of the sentence is encoded into is

\[7 + 9 + 7 + 6 = 29.\]
4. Consider a Lempel-Ziv compression algorithm with a window size of 16. Suppose the encoder has already encoded the following text:

```
MY_BROTHER_TOMATE
```

and is left to encode the remainder of the sentence which reads as

```
A_TOMATO
```

Indicate the sequence of phrases that are encoded. Count the number of bits the remainder of the sentence is encoded into.

<table>
<thead>
<tr>
<th>Code</th>
<th># of bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 3, 2)</td>
<td>1 + 4 + 2</td>
</tr>
<tr>
<td>(1, 9, 4)</td>
<td>1 + 4 + 4</td>
</tr>
<tr>
<td>(1, 8, 2)</td>
<td>1 + 4 + 2</td>
</tr>
<tr>
<td>(1, 3, 1)</td>
<td>1 + 4 + 1</td>
</tr>
</tbody>
</table>

The number of bits the remainder of the sentence is encoded into is

\[7 + 9 + 7 + 6 = 29.\]