A1  a) The source has 7 symbols. The minimum worst-case length of a variable length code is attained by a code which assigns symbols \{∅, 0, 1, 00, 01, 10, 11\} to symbols with decreasing probabilities. Thus, \(L_0(X) = 2\).

In fact, the same assignment as above also attains the minimum average codeword length. Thus,

\[
L_0(X) = 0 \times 1/2 + 2 \times 1 \times 1/8 + 4 \times 2 \times 1/16 = 3/4.
\]

Note that since \(-\log p(x)\) is an integer for every \(x\), \(L_0^{\text{unique}}(X) = L_0^{\text{prefix}}(X) = H(X) = 9/4\) and can be attained by the Huffman code or the Shannon-Fano code (see, also, Q2(a)).

b) In the manner of \(L_0(X)\) and \(L_0^{\epsilon}(X)\), \(L_\epsilon(X)\) and \(L_\epsilon^{\epsilon}(X)\) are also attained by the code which assigns shortest available codewords in decreasing order of probabilities to the symbols till the probability adds up to greater than 1 \(-\epsilon\); all the remaining symbols are assigned to the shortest codeword, the empty sequence. Thus, the optimal lengths for \(\epsilon = 0.25\) are attained by the following code:

<table>
<thead>
<tr>
<th>symbol</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>∅</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>{3, 4, 5, 6}</td>
<td>∅</td>
</tr>
</tbody>
</table>

Thus, \(L_{0.25}(X) = 1\) and \(L_{0.25}^{\epsilon}(X) = 1/4\).

A2  a) As shown in the class, if \(-\log P(x)\) is an integer for every \(x\) then there Shannon exists a prefix-free code which assigns the length \(l(x) = -\log P(x)\) to every symbol \(x\). Clearly, the average length of this code is \(-\sum_x P(x) \log P(x) = H(X)\), which further implies \(L_0^{\text{prefix}}(X) = H(X)\) since \(L_0^{\text{prefix}}(X) \geq H(X)\).

For the converse, note that \(L_0^{\text{prefix}}(X) = H(X)\) only if the Huffman code has average length \(H(X)\). Let \(l(x)\) be the length of codeword assigned to symbol \(x\) by the Huffman code. Thus, using log-sum inequality and Kraft’s inequality,

\[
\sum_x P(x) l(x) - H(P) = \sum_x P(x) \log \frac{P(x)}{2^{-l(x)}} \\
\geq -\log \sum_{x'} 2^{-l(x')} \\
\geq 0,
\]
with equality if and only if $\sum_{x'} 2^{-l(x')} = 1$ and
\[
P(x) = \frac{2^{-l(x)}}{\sum_{x'} 2^{-l(x')}} = 2^{-l(x)}, \quad x \in \mathcal{X}.
\]
Therefore, for every $x \in \mathcal{X}$, $-\log P(x) = l(x)$, an integer.

b) In view of the result in part (a), $H_0^{\text{prefix}}(X) > H(X)$ if there is a symbol $x$ where $-\log P(x)$ is not an integer. Consider the following example with $\mathcal{X} = \{0, 1, 2\}$:

\[
P_X(0) = 0.5, \quad P_X(1) = 0.5 - P_X(2) = 0.2.
\]

The Huffman code for this example has the following codeword assignment

<table>
<thead>
<tr>
<th>symbol</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
</tbody>
</table>

Since Huffman code is optimal and the average length of the Huffman code equals $3/2$,
\[
H_0^{\text{prefix}}(X) = 3/2.
\]

On the other hand
\[
H(X) = 1/2 + 2/10 \log 10/2 + 3/10 \log 10/3
\]
\[
= 1 + 1/2(2/5 \log 5/2 + 3/5 \log 5/3)
\]
\[
= 1 + (1/2) h(2/5)
\]
\[
< 3/2.
\]

A3 a) Construction of Huffman code.

(a) Assign symbols 2 and 3 as the left and the right child of a node $(2, 3)$. We now have three symbols, 1, $(2, 3)$, and 4, each with the same probability $1/3$.

(b) Assign symbols 1 and $(2, 3)$ as the left and the right child of a node $(1, (2, 3))$. We now have two symbols, $(1, (2, 3))$ and 4, the first with probability $2/3$ and the second with probability $1/3$.

(c) Assign $(1, (2, 3))$ and 4 as left and right child of the root.

The Huffman code is obtained by moving from root to the leaves and associating each left child with 0 and the right child with 1. We get the following codewords

<table>
<thead>
<tr>
<th>symbol</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>
Construction of Shannon-Fano code. Let \( x_1 = 1, x_2 = 4, x_3 = 2, x_4 = 3 \) and let \( p_i = P(x_i) \), \( i = 1, 2, 3, 4 \). Thus, \((p_1, p_2, p_3, p_4)\) is a decreasing sequence. Let \( F_1 = 0, F_2 = p_1 = 1/3, F_3 = p_1 + p_2 = 2/3, F_4 = F_3 + 1/6 = 5/6 \). The codeword for the symbol \( x_i \) is obtained by taking the \( \lceil -\log p_i \rceil \) most significant bits of binary representation of \( F_i \). Thus, \( x_1 \) and \( x_2 \) is represented using 2 bits each and \( x_3 \) and \( x_4 \) using 3 bits each. We get the following code

<table>
<thead>
<tr>
<th>symbol</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 = 1 )</td>
<td>00</td>
</tr>
<tr>
<td>( x_2 = 4 )</td>
<td>01</td>
</tr>
<tr>
<td>( x_3 = 2 )</td>
<td>101</td>
</tr>
<tr>
<td>( x_4 = 3 )</td>
<td>110</td>
</tr>
</tbody>
</table>

Construction of Elias code. We work with the original ordering of the symbols. Then,

\[
\begin{align*}
F(1) &= 0 + P(1)/2 = 1/6, \\
F(2) &= P(1) + P(2)/2 = 5/12, \\
F(3) &= P(1) + P(2) + P(3)/2 = 7/12, \\
F(4) &= P(1) + P(2) + P(3) + P(4)/2 = 5/6.
\end{align*}
\]

The Elias code is obtained by representing the symbol \( i \) with the \( \lceil -\log P(i) \rceil + 1 \) most significant bits in the binary representation of \( F(i) \). Thus, we represent \( 1, 2, 3, \) and \( 4 \) with codewords of length \( 3, 4, 4, \) and \( 3 \), respectively. We get the following code

<table>
<thead>
<tr>
<th>symbol</th>
<th>codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>000</td>
</tr>
<tr>
<td>2</td>
<td>0110</td>
</tr>
<tr>
<td>3</td>
<td>1001</td>
</tr>
<tr>
<td>4</td>
<td>110</td>
</tr>
</tbody>
</table>

b) Both Huffman code and Shannon-Fano code depends on ordering the elements of \( \mathcal{X} \) in decreasing order of probabilities, but this ordering is not unique since there are multiple elements with the same probability. Thus, none of these constructions is unique. Similarly, the Elias code too is not unique since it depends on (an arbitrary) ordering of the elements of \( \mathcal{X} \).

c) We will give the implementation of the arithmetic coding with infinite precision. Thus, we will compute the limits of intervals as real numbers. At each step we will maintain the starting point and the width of the interval.

Encoding:

(a) \( X_1 = 3 \Rightarrow C_1 = F(X_1) = 1/2, W_1 = P(X_1) = 1/6 \).

(b) \( X_2 = 1 \Rightarrow C_2 = C_1 + W_1 \times F(X_2) = 1/2, W_2 = W_1 \times P(X_2) = 1/18 \).

(c) \( X_3 = 4 \Rightarrow C_3 = C_2 + W_2 \times F(X_3) = 29/54, W_3 = W_2 \times P(X_3) = 1/54 \).

(d) \( X_4 = 2 \Rightarrow C_4 = C_3 + W_3 \times F(X_4) = 88/162, W_4 = W_3 \times P(X_4) = 1/324 \).

Thus, we can encode the sequence 3142 by the shortest binary sequence such that the real number in [0, 1] represented by the sequence belongs to the interval \([176/324, 177/324) = [0.543209..., 0.54629...]; it suffices to find a binary number in the interval [0.544, 0.546]. Clearly,
the first bit is 1. For the next three bits, consider $2 \ast [0.044, 0.046] = [0.088, 0.092]$, $4 \ast [0.044, 0.046] = [0.176, 0.184]$, and $8 \ast [0.044, 0.046] = [0.352, 0.368]$. Thus, the second, the third and the fourth bits are 0. For the fifth bit, consider $2 \ast [0.352, 0.368] = [0.704, 0.736]$. Thus, the fifth bit is 1. For the sixth bit, consider $2 \ast [0.204, 0.236] = [0.408, 0.472]$. Thus, sixth bit is also 0. For the seventh bit, consider $2 \ast [0.316, 0.444] = [0.632, 0.888]$. Thus, the seventh bit is 1. For the eighth bit, consider $2 \ast [0.132, 0.388] = [0.264, 0.776]$, which contains 1/2. Thus, the ninth bit is 1 and we can stop. The resulting codeword is 100010111.

Decoding: The received codeword 100010111 corresponds to the number $c = 0.544921875$.

(a) Let $c_1 = c$. Since $c_1 \in [F(3), F(4)] = [1/2, 1/2 + 1/6]$, $X_1 = 3$.

(b) Consider $c_2 = (c_1 - F(3))/P(3) = 0.26953125$. Since $c_2 \in [F(1), F(2)] = [0, 1/3)$, $X_2 = 1$.

(c) Consider $c_3 = (c_2 - F(1))/P(1) = 0.80859375$. Since $c_3 \in [F(4), 1] = [2/3, 1)$, $X_3 = 4$.

(d) Consider $c_4 = (c_3 - F(4))/P(4) = 0.42578125$. Since $c_3 \in [F(2), 1] = [1/3, 1/2)$, $X_3 = 2$.

Thus, the first encoded symbol was decoded first and the FIFO property holds.

A4 Consider encoder $f_1(x)$ which encodes the type of the sequence $x$ and the encoder $f_2(x)$ which encodes the enumeration of $x$ in the type class $T_{P_X}$. Many choices of $f_1$ and $f_2$ are possible. We choose both $f_1$ and $f_2$ to be fixed length codes, first of length $l_1 = \lceil \log |T| \rceil$ and the second of length $l_2(P_x) = \lceil \log |T_{P_x}| \rceil$.

a) The answer to this question depends on your specific choice of $f_1$ and $f_2$. For the choice above, since $f_1$ is a block code, the codewords $c(x)$ and $c(x')$ corresponding to sequences $x$ and $x'$ of different types have different first $l_1$ bits and therefore one can’t be a prefix of the other. If $x$ and $x'$ are of the same type $Q$, both have the same length and with different last $l_2 = \lceil \log |T_Q| \rceil$ bits. Thus, in this case too, no codeword can be a prefix of the other, which implies that the code above is prefix-free (and hence is uniquely decodeable and nonsingular).

b) For a sequence $x$, the length $l(x)$ of the codeword $c(x)$ is bounded above as

$$l(x) \leq l_1 + l_2(x) \leq |X| \log(n + 1) + nH(P_X) + 2,$$

where we have used the following facts:

$$|T| \leq (n + 1)^{|X|},$$

$$|T_Q| \leq 2^{nH(Q)}.$$

Thus, denoting by $Q(X^n)$ the type of the random sequence $X^n$ generated by $P^n$, the average length $\overline{T}_n(P)$ is bounded above as

$$\overline{T}_n(P) \leq |X| \log(n + 1) + n\mathbb{E}H(Q(X^n)) + 2$$

$$\leq |X| \log(n + 1) + nH(\mathbb{E}Q(X^n)) + 2$$

$$= |X| \log(n + 1) + nH(P) + 2,$$

where the second inequality uses Jensen’s inequality and the equality uses $\mathbb{E}Q(X^n) = P$.

Thus, for every$^1$ $P$

$$\overline{T}_n(P) - nH(P) \leq |X| \log(n + 1) + 2,$$

$^1$If instead of the blockcode $f_1$, we used an appropriate prefix-free code of variable length, it can be shown that $\overline{T}_n(P) - nH(P) \leq \frac{|X| - 1}{2} \log n + \text{constant}$. 

4
and so,

$$\lim_{n \to \infty} \frac{1}{n} [I_n(P) - nH(P)] = 0.$$  

Therefore,

$$\lim_{n \to \infty} \frac{1}{n} I_n(P) = H(P)$$

for every pmf $P$. 