1. (9 pts) **Codeword Lengths.**

Consider a lossless compression code that uses five binary codewords with lengths \{1, 2, 3, 4, 5\}.

(a) Prove that a uniquely decodable code exists with these codeword lengths. While you are at it, prove that a prefix free code exists with these codeword lengths.

(b) Exhibit a prefix free code with these lengths.

(c) Either give a PMF for which your code is a Huffman code, or explain why your code cannot be a Huffman code for any PMF.

2. (8 pts) **Code Analysis.**

Consider a binary source code that employs the following codewords:

```
0
01
001
001001
001001001
```

(a) (4 pts) Is this code instantaneous? Does an instantaneous binary source code exist with these codeword lengths? (Justify your answers.)

(b) (4 pts) Is this code uniquely decodeable? Does a uniquely decodeable binary source code exist with these codeword lengths? (Justify your answers.)
3. (8 pts) **Reversal.**

Consider a binary source code that employs the following codewords:

0  
01  
011  
0111  
01111  
011111  
111111

(a) (4 pts) Is this code instantaneous? Does an instantaneous binary source code exist with these codeword lengths? (Justify your answers.)

(b) (4 pts) Is this code uniquely decodeable? Does a uniquely decodeable binary source code exist with these codeword lengths? (Justify your answers.)

4. (8 pts) **Slackness in the Kraft inequality.**

An instantaneous code has word lengths \( l_1, l_2, \ldots, l_m \) which satisfy the strict inequality

\[
\sum_{i=1}^{m} D^{-l_i} < 1.
\]

The code alphabet is \( D = \{0, 1, 2, \ldots, D - 1\} \). Show that there exist sequences of code symbols in \( D^* \) which cannot be decoded into sequences of codewords. \( D^* \) is the set of finite length strings of symbols from the code alphabet.

5. (8 pts) **Huffman Code.**

\( X \) is distributed according to the following probability mass function:

\[
\frac{1}{2} \quad \frac{1}{4} \quad \frac{1}{8} \quad \frac{1}{16} \quad \frac{1}{32} \quad \frac{1}{64} \quad \frac{1}{64}.
\]

(1)

Find a binary Huffman code for this distribution. Does it achieve the corresponding entropy limit on compression?
6. (12 pts) Non-binary Huffman Code?

$X$ is distributed according to the following probability mass function:

\[
\begin{array}{cccc}
\frac{1}{3} & \frac{1}{3} & \frac{1}{9} & \frac{1}{27} & \frac{1}{27} & \frac{1}{27}.
\end{array}
\]

(a) (5 pts) Find a binary Huffman code for $X$. Compute its average length in bits.

(b) (5 pts) Find a ternary Huffman code (for which there are three symbols instead of two, three branches in every Huffman step instead of two.) Compute its average length (average number of ternary symbols, which are sometimes called trits).

(c) (2 pts) Does the ternary Huffman code achieve the corresponding ternary entropy limit on compression? Which code is more efficient in this case, the ternary or the binary? Explain your answer.

7. (9 pts) Bad codes. Which of these codes cannot be Huffman codes for any probability assignment?

(a) \{0, 10, 11\}.

(b) \{00, 01, 10, 110\}.

(c) \{01, 10\}.

8. (10 pts) Simple optimum compression of a Markov source. Consider the 3-state Markov process having transition matrix

\[
U_{n-1} \begin{bmatrix}
S_1 \\
S_2 \\
S_3
\end{bmatrix}
= \begin{bmatrix}
\frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\
0 & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}
\]

Thus the probability that $S_1$ follows $S_3$ is equal to zero. Design 3 codes $C_1, C_2, C_3$ (one for each state $S_1, S_2, S_3$), each code mapping elements of the set of $S_i$'s into sequences of 0's and 1's, such that this Markov process can be sent with maximal compression by the following scheme:

(a) Note the present symbol $S_i$.

(b) Select code $C_i$.

(c) Note the next symbol $S_j$ and send the codeword in $C_i$ corresponding to $S_j$.

(d) Repeat for the next symbol.

What is the average message length of the next symbol conditioned on the previous state $S = S_i$ using this coding scheme? What is the unconditional average number of bits per source symbol? Relate this to the entropy rate $\mathcal{H}$ of the Markov chain.
9. (18 pts) *Arithmetic Coding.*

In this problem you will encode a sequence using arithmetic coding and decode a different sequence using arithmetic coding.

The following probability model should be used for both encoding and decoding. Note that \(\langle eot\rangle\) can only occur as the last character in the string.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Probability</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>0.4</td>
<td>[0.0, 0.4)</td>
</tr>
<tr>
<td>(b)</td>
<td>0.35</td>
<td>[0.4, 0.75)</td>
</tr>
<tr>
<td>(c)</td>
<td>0.15</td>
<td>[0.75, 0.9)</td>
</tr>
<tr>
<td>(\langle eot\rangle)</td>
<td>0.1</td>
<td>[0.9, 1)</td>
</tr>
</tbody>
</table>

(a) Encode \(caab\langle eot\rangle\) using arithmetic coding. Use the smallest number of bits possible assuming that all bits to the right of the last transmitted bit are zeros.

(b) Decode 0011010111. Assume that all bits to the right of the last transmitted bit are zeros.

The MATLAB functions `binary2real`, `encode_symbol`, and `send_bit` discussed in lecture are available on the course web site. Note that the command `format long` will cause MATLAB to print out 14 decimal places instead of 4. You should feel free to enhance these routines to print out more information.

You will need to write your own function `decode_symbol`.