Complete all problems. You are not permitted to look at solutions of previous year assignments. You can work together in groups, but all solutions have to be written up individually.

**Problem 1: Conditional Probabilities (10pt)**

Given the following conditional probabilities for a two state Markov Chain what factor would one save by using the conditional entropy instead of the unconditional entropy?

\[
\begin{align*}
p(w|w) &= 0.95 \\
p(b|w) &= 0.05 \\
p(w|b) &= 0.2 \\
p(b|b) &= 0.8
\end{align*}
\]

**Problem 2: Arithmetic Codes (10pt)**

Given the following probability model:

<table>
<thead>
<tr>
<th>Letter</th>
<th>(p(a_i))</th>
<th>(f(a_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0.1</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>c</td>
<td>0.7</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Decode the 4 letter message given by 00011011010 assuming it was coded using arithmetic coding. Why is this message longer than if we simply had used a fixed-length code of 2 bits per letter, even though the entropy of the set \{0.1, 0.2, 0.7\} is just a little more than 1 bit per letter. Note: once you figure out how to do the decoding, it should not take more than five minutes on a calculator or scripting language.

**Problem 3: Decoding Prefix Codes (20pt)**

Being able to quickly decode prefix codes is extremely important in many applications. Assume you have a machine with word length \(w\) (e.g. 32 bits). Assume you are give some prefix code for a message set, and the longest codeword is \(w/2\) bits (or less). Assume that a sequence of codes is stored in memory broken into words (i.e. the first \(w\) bits are in the first word, etc.).

The naive way to decode is to use a binary tree and take constant time per bit by traversing the tree. Describe how to decode each codeword in constant time (independently of \(w\)). Hint, you can use \(O(2^w/2)\) preprocessing time, and the same amount of memory. Please don’t use more than half a page to describe the method.
Problem 4: Bounds on Prefix Codes (20pt)

A. Prove the first part of the Kraft-McMillan inequality for Prefix Codes. In particular show that for any prefix code $C$,
\[ \sum_{(s,w) \in C} 2^{-l(w)} \leq 1. \]

B. Prove that if you have $n = 2^k$ codewords in a prefix code and that if one of them is shorter than $k$ bits, then at least two must be longer than $k$ bits.