Algorithms in the Real World (15-853), Fall 04
Assignment #1

Complete 70 points.
Due October 4.

Problem 1: Conditional Probabilties (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?

$$p(w|w) = .9 \quad p(b|w) = .1$$
$$p(w|b) = .2 \quad p(b|b) = .8$$

Problem 2: Uniquely Decipherable Codes (10pt)
Devise a uniquely decipherable code that is not a prefix code.

Problem 3: Arithmetic Codes (10pt)
Given the following probability model:

Letter	$p(a_i)$	$f(a_i)$
a	.1	0
b	.2	.1
c	.7	.3

Decode the 4 letter message given by `01001110110` 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 $\{.1, .2, .7\}$ 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 4: Decoding Prefix Codes (20pt)
Assume you have a machine with word length w (e.g. 32 bits). Assume you are give some prefix code for encoding the numbers 1–256 for which 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). You can use an array of size $2^{w/2}$ (e.g. 65536 entries on a 32-bit machine).
Please don't use more than a page to describe the method

Problem 5: 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.

Problem 6: PPM, LPZ and BW (20pt)

The string `bcabccabc` is encoded using PPM. The (partial) dictionary constructed during encoding is given below. For the following questions, assume that escape count is given by the number of different characters for each context, and exclusion is *not* used, unless specified. Assume that the alphabet has 26 characters, and use $k = 2$.

(a) Fill in the empty spaces in the dictionary below.

Context	Counts	Context	Counts	Context	Counts
empty	$a = 2$	a	$b = 2$	ab	$c = 2$
	$b = 3$				
	$c = 4$	b			
		c			

Figure 1: Dictionary

(b) Suppose the next letter in the string is `b`. Compute the number of bits required to encode `b`, and also list the changes made to the dictionary. (You do not have to compute the exact number of bits, simply write it as an expression containing logs).

(c) Now assume that exclusion is used. Recompute the number of bits required to encode the character `b`.

(d) Encode the above string `bcabccabc` using LZ77, with an unbounded lookahead buffer, and a window of size 4.

(e) Encode the above string using Burrows-Wheeler. Just show the sequence of characters after the BW transform (don't bother compressing using move to front).