Algorithms in the Real World
Assignment #1 Due Date: 12 Sept 97

Collaboration policy: You are welcome to talk about this or other assignments with others and even discuss problems using a whiteboard or scratch paper. You are not allowed, however, to show or give any work that you did on your own pertaining to the assignment to someone else.

Problem 1: Assume an alphabet $S = \{a, b, c\}$ has the probabilities

$$p_a = .2 \quad p_b = .7 \quad p_c = .1$$

(a) What is the entropy $H(S)$?

(b) What is the average length of the Huffman code for S (remember to weigh the average based on the probabilities).

(c) What is the average length of the Huffman code if you group characters into pairs.

(d) For a long string what is the average per-character length of an arithmetic code for S (assume the asymptotic value and that the message abides by the probabilities).

(d) For a long string what is the average per-character length of an arithmetic code for S assuming your model are the probabilities given above, but you are coding a message which consists of just c's (i.e., your data does not fit your model).

(e) Given the following conditional probabilities what is the conditional entropy (this is the average per-character length that would be achieved by a good code that takes advantage of the probabilities).

$$\begin{array}{lll} p(a|a) = .25 & p(b|a) = .65 & p(c|a) = .1 \\ p(a|b) = .1 & p(b|b) = .8 & p(c|b) = .1 \\ p(a|c) = .8 & p(b|c) = .1 & p(c|c) = .1 \end{array}$$

The notation $p(x|y)$ is the probability of x given that the previous character is y.

Problem 2: Prove that the average length $l_a = \sum p_i l_i$ for a Huffman code for a set of messages S is bounded by the inequality $l_a \leq H(S) + 1$. Note that given a maximum probability p_m a tigher upper bound of

$$l_a < \begin{cases} H(S) + p_m & p_m < .5 \\ H(S) + p_m + .086 & p_m \geq .5 \end{cases}$$

can be shown. Feel free to prove this stronger result.

Problem 3: Assume we are coding a 4 letter alphabet with LZW and we start with the dictionary containing the 4 letters $\{a, b, c, d\}$ in locations 0, 1, 2 and 3. Entries are added to the dictionary in sequential order. Decode the following message that was coded with LZW.

```
0 0 1 2 4 7
```

Problem 4: Using the LZSS (the variant of LZ77 described in the reading "Introduction to data compression" by Peter Gutmann) and assuming a window size W and a lookahead buffer size B, how many code words will it take to code a message of length L that contains all the same character. Assume each ¡offset,size¿ pair or character is a codeword. Also assume the window is initialized to some other character.

Problem 5: Lets say we had the following sequence of values

```
10 11 12 11 12 13 12 11 10 -10 8 -7 8 -8 7 -7
```

(a) Would we would get better compression by doing a cosine transform across the whole group of 16 or by blocking the 16 entries into two groups of 8 and transforming each separately? Explain briefly?

(b) If using a wavelet transform are we better off blocking the entries into two groups of 8?