# CSE 490 G
# Assignment 3
Due Friday, January 27, 2006

1. Consider the probability distribution $a : 1/4, b : 1/2, c : 1/4$.
   (a) Use arithmetic coding with scaling to code the string *bbbbba*. Show the steps in the process and the value of $C$ which keeps track of the number of complementary bits to be output after a 0 or 1 is output. I chose this example because the scaled interval are very easy to calculate.
   (b) Use arithmetic decoding with scaling to decode 00000000001 (10 zeros followed by a 1) assuming the string decoded is of length 6.

2. In some situations a data file has the property of having a relatively small "working set". This means that the current symbol most often comes from a fairly small set of symbols. For example, consider the string $x$ of symbols in he alphabet $\{a, b, c, d, e, f\}$:

   $x =$ `abccaabbbcabddcbcbddceddeddeccdeefeffddefdfdeeff`

   which tends to have a working set of about size 3.

   In the move-to-front algorithms we first give an initial index to each symbol as follows:

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| a | b | c | d | e | f |

   Suppose symbol $x$ with index $i$ is encountered in the input stream. The index $i$ is output. Then the index of $x$ becomes 0 and all the symbols indexed $< i$ have their index increased by 1. For example the input $y$

   $y =$ `bbbfbb`

   has output 100510 because after the first `b` is input the indexing becomes

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| b | a | c | d | e | f |

   and after the `f` is input then the indexing becomes

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| f | b | a | c | d | e |

and after the fourth `b` is input the indexing becomes

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| b | f | a | c | d | e |

(a) Compute the empirical entropy of the string $x$. (The empirical entropy is done using the frequencies of the symbols found in the string.)

(b) Compute the empirical entropy of the string output in the move-to-front algorithm executed on $x$.

(c) In move-to-front compression both the encoder and decoder know the initial indexing and the output of the move-to-front algorithm is losslessly encoded, say with arithmetic coding. Give one example of a data set that might be amenable to move-to-front compression, and explain why it is so. English text is not an example.