

CMPS340 Fall 2008
HW #5: Huffman Coding
Due: 12noon, Friday, December 5

1. Suppose that we have a file that is to be compressed. Suppose, further, that we decide to view it as being composed of 3-bit blocks, interpreting each such block as being a single character. (For convenience, we assume that the length of the file, in bits, is divisible by three.) Under this interpretation, there are eight characters in the *source alphabet*, one for each of the distinct bit strings of length three. We use A to name this alphabet. The frequency with which each member of A occurs in the file is given in the following table. (For convenience, we refer to the eight characters as a_0, a_1, \dots, a_7 , where a_k corresponds to the length 3 binary numeral that represents k .)

Character	Frequency	Character	Frequency
a_0 (000)	17%	a_4 (100)	8%
a_1 (001)	39%	a_5 (101)	18%
a_2 (010)	4%	a_6 (110)	3%
a_3 (011)	6%	a_7 (111)	5%

(a) Construct a Huffman tree corresponding to these frequencies.

Note: In order to ensure that the correct answers to parts (b) and (c) below are unique, follow this rule in labeling the edges of the Huffman tree: From each non-leaf node there are edges connecting it to its two children, one labeled 0 and the other labeled 1. Use 0 to label the edge that goes to the subtree having the leaf associated to the “alphabetically smallest” character. For example, if a node’s two children are the roots of subtrees whose “leaf sets” are $\{a_2, a_3, a_6\}$ and $\{a_1, a_5\}$, respectively, then the edge into the latter should be labeled 0, because a_1 is in the latter and is the smallest among all characters in the two subtrees. **End of note.**

(b) Show the Huffman code (i.e., the mapping from A to binary codewords) implied by your answer to (a).

(c) Letting c_i be the codeword for a_i and l_i be the unary representation for $|c_i|$ (the length of c_i), suppose that the codeword table (which should be written at the beginning of the compressed version of the file) is encoded as $l_0c_0l_1c_1 \dots l_7c_7$. Show the corresponding bit string, and annotate it to show the boundaries between adjacent elements. (Recall that the unary representation for $k > 0$ is $0^{k-1}1$.)

(d) Compute the ratio between the lengths of the compressed version of the file and the original file. Assume that the length of the original file is 24000 bits. Show your work. Keep in mind that the number of bits used, on average, for encoding a character in the compressed version of the file is the sum

$$\sum_{0 \leq i < 8} (|c_i| \cdot freq(a_i))$$

where $freq(a_i)$ is the frequency with which a_i occurs in the original file. Also recall that the compressed version of the file contains a representation of the codeword table (as described in (c)).

(e) Now suppose that we had assigned codewords to the characters in the source alphabet in accord with **canonical** Huffman coding. (Follow the rule that if a_i and a_j have codewords of the same length and $i < j$, then the codeword assigned to a_i should be numerically/lexicographically smaller than the codeword assigned to a_j .) Show the canonical mapping from A to codewords.

2. The set of strings $L = \{aa, aabab, abb, ba, bb\}$ is *not* uniquely decipherable (u.d.). Demonstrate this by showing two factorizations of any string of your choosing. To receive full credit, the string you choose must be a shortest string having two factorizations.