1. The optimality of the Huffman code is discussed in detail in Chapter 5.8 of your text book. Go through this instructive exercise carefully. In the next homework we will work on an extension of this line of thought. In this exercise we ask you to find the binary Huffman code for the source with probabilities \((\frac{1}{3}, \frac{1}{5}, \frac{1}{5}, \frac{2}{15}, \frac{2}{15})\). Argue that this code is also optimal for the source with probabilities \((\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})\).

2. Consider the random variable \(X\) that takes values \(x_1, \ldots, x_7\) with probabilities \(p_i = P(X = x_i)\) as follows: \(p_1 = 0.49, p_2 = 0.26, p_3 = 0.12, p_4 = 0.04, p_5 = 0.04, p_6 = 0.03, p_7 = 0.02\).

   (a) Find a binary Huffman code for \(X\).
   (b) Find the expected code length for this encoding.
   (c) Find a ternary Huffman code for \(X\).

3. In lecture we found that the optimal average codeword length is less than \(H + 1\) where \(H\) is the entropy of the source. Give an example of a random variable whose expected length of the optimal code is arbitrarily close to \(H + 1\). In other words, for any fixed \(\varepsilon > 0\) construct a source probability distribution for which the optimal code has average length \(L > H + 1 - \varepsilon\).

4. Player A chooses some object in the universe, and player B attempts to identify the object with a series of yes-no questions. Suppose that player B is clever enough to use the code achieving the minimal expected length with respect to player A’s distribution. We observe that player B requires an average of 38.5 questions to determine the object. Find a rough lower bound to the number of objects in the universe.

5. A code is not uniquely decodable if and only if there exists a finite sequence of code symbols which can be resolved into sequences of codewords in two different ways; for example the codeword sequence could be split as either \((A_1 A_2 A_3 \ldots A_m)\) or \((B_1 B_2 \ldots B_n)\) where each \(A_i\) and \(B_j\) is a codeword. If the length of \(A_1\) is larger than that of \(B_1\) (without loss of generality), then \(B_1\) must be a prefix of \(A_1\) with some “dangling suffix” left over in \(A_1\). Each dangling suffix must in turn be either a prefix of a codeword (in this case the dangling suffix of \(A_1\) is a prefix of \(B_2\), assuming that the length of \(B_1 B_2\) is more than the length of \(A_1\)) or have another codeword as its prefix, resulting in another dangling suffix. Finally, the last dangling suffix in the sequence must also be
a codeword. Thus one can set up a test for unique decidability – this is the so-called Sardinas-Patterson test – in the following way. Construct a set $S$ of all possible dangling suffixes. The code is uniquely decodable if and only if $S$ contains no codeword.

(a) State the precise rules for building the set $S$.

(b) Suppose the codeword lengths are $\ell_i, i = 1 \ldots m$. Find a good upper bound on the number of elements in the set $S$.

(c) Determine which of the following codes is uniquely decodable.

i. $\{0, 10, 11\}$
ii. $\{0, 01, 11\}$
iii. $\{0, 01, 10\}$
iv. $\{0, 01\}$
v. $\{00, 01, 10, 11\}$
vi. $\{110, 11, 10\}$
vii. $\{110, 11, 100, 00, 10\}$