Solutions: Homework Set # 1

Problem 1

By definition

$$L_1 = \min_{\text{instantaneous code}} \sum_{i=1}^{m} p_i l_i^{100},$$

and

$$L_2 = \min_{\text{uniquely decodable code}} \sum_{i=1}^{m} p_i l_i^{100}.$$

Since all instantaneous codes are uniquely decodable, the domain of the first minimization problem is a subset of that of the second one. Therefore, $L_1 \geq L_2$.

Assume $\mathcal{C}$ with codeword lengths $l_1, l_2, \ldots, l_m$ be a uniquely decodable code which minimizes L. Since all uniquely decodable codes satisfy the Kraft's inequality (McMillan's theorem), we know that

$$\sum_{i=1}^{m} D^{-l_i} \leq 1.$$

Hence, there exist an instantaneous code $\mathcal{C}'$ with the same set of codeword lengths, and therefore the same L. It is clear that the best solution for the first optimization problem cannot exceed L, and $L_1 \leq L_2$.

Summarizing the two inequalities, we obtain $L_1 = L_2$.

Remark. Note that although Huffman code is optimal for minimizing $L(1) = \sum_{i=1}^{m} p_i l_i$, but they can be arbitrary far from the optimal code if want to minimize $L(100) = \sum_{i=1}^{m} p_i l_i^{100}$. For example, assume we want to design a code for a source with four symbols (a, b, c, d) with probability distribution $(p_a, p_b, p_c, p_d) = (\frac{3}{4}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12})$. A binary Huffman code for this source would be $(0, 10, 110, 111)$ with codeword lengths $(1, 2, 3, 3)$, and $L(1) = \frac{17}{12}$ and $L(100) \simeq 8.6 \times 10^{46}$. However, we can design another code $(00, 01, 10, 11)$ with codeword lengths $(2, 2, 2, 2)$ and $L(1) = 2$, but $L(100) = 1.3 \times 10^{30}$, which is much smaller than $L(100)$ of the Huffman code. This example shows that depending on the objective function (what we want to be minimized), different codes might be optimal.

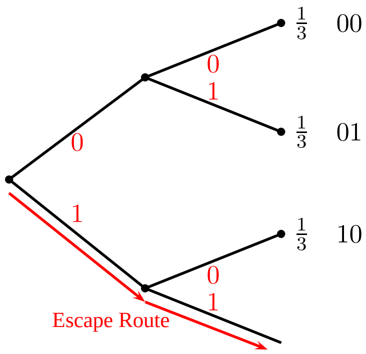
Problem 2

Let $l_{\max} = \max\{l_1, l_2, \cdots, l_m\}$. There are $D^{l_{\max}}$ number of sequences with length $l_{\max}$. Of these sequences, if you fix the beginning to be i^{th} codeword, you will have $D^{l_{\max}-l_i}$ sequences, in other words, you have $D^{l_{\max}-l_i}$ sequences starting with the i^{th} codeword. Since we have a prefix-free code, no two codewords can start with the same codeword, so the total number of sequences which start with some codeword is

$$\sum_{i=1}^{m} D^{l_{\max}-l_i} = D^{l_{\max}} \sum_{i=1}^{m} D^{-l_i} < D^{l_{\max}} \tag{1}$$

Hence there exist some sequences which do not start with any codewords. These sequences cannot be decoded.

alternatively we can say that if the Kraft inequality is not satisfied with equality, there are some leaves in the tree that are not assigned to codewords and from these leaves one can find a '*escape route*' for the sequences which cannot be decoded. If one looks carefully to these two solutions, can conclude that they are actually equivalent. This can be seen in the following tree representation in the case of $D = 2$.



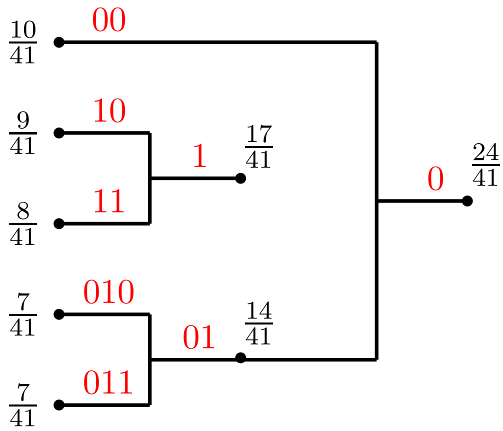
Problem 3

The values of $l_i{}^*$ that minimize $\sum p_i l_i{}^*$ are $l_i{}^* = -\log p_i$ To obtain an optimal prefix free code we use the Huffman procedure, which results in lengths $(l_1{}^H, l_2{}^H \ldots)$

(a)

$$(l_1{}^*, l_2{}^*, l_3{}^*, l_4{}^*, l_5{}^*) = \left(\log \frac{41}{10}, \log \frac{41}{9}, \log \frac{41}{8}, \log \frac{41}{7}, \log \frac{41}{7}\right) = (2.04, 2.19, 2.35, 2.55, 2.55)\,. \tag{2}$$

Huffman procedure:



$$(l_1{}^H, l_2{}^H, l_3{}^H, l_4{}^H, l_5{}^H) = (2, 2, 2, 3, 3)\,. \tag{3}$$

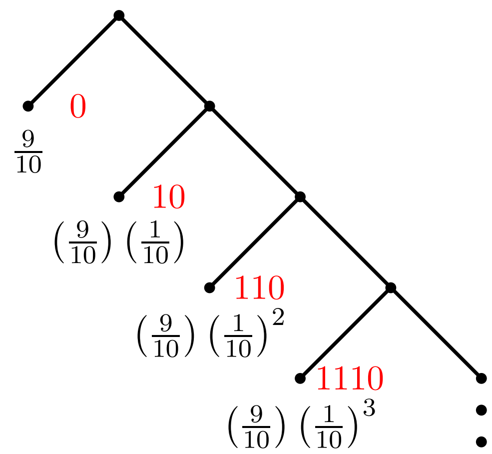
(b)

$$p_i = \left(\frac{9}{10}\right)\left(\frac{1}{10}\right)^{i-1}. \tag{4}$$

$$l_i{}^* = -\log \left(\frac{9}{10}\right)\left(\frac{1}{10}\right)^{i-1} = \log \left(\frac{10}{9}\right) + (i-1)\log 10. \tag{5}$$

$$(l_1{}^*, l_2{}^*, \dots) = (0.152, 3.47, 6.8, 10.12, \dots)\,. \tag{6}$$

Huffman procedure cannot be directly applied, due to the infinite nature of the code. However, one can quickly realize that in the optimal code, codeword i is $\underbrace{111\dots1}_{i-1}0$.



$$\left(l_1{}^H, l_2{}^H, \dots\right) = (1, 2, 3, 4, \dots)\,. \tag{7}$$

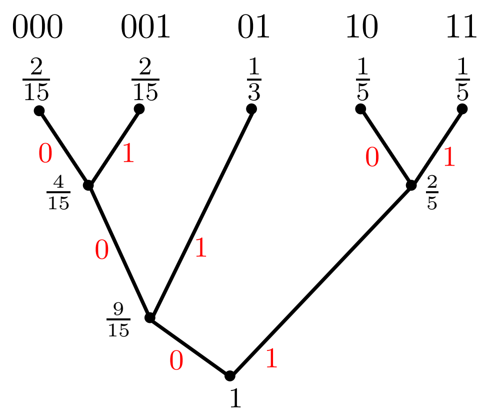
Problem 4

(a) For any received sequence, work from the end with the inverse of the suffix code. The inverted codewords satisfy the prefix-free condition, hence the code is uniquely decodable.

(b) Similarly to part (a), start by reversing the codewords. This operation does not change the codeword lengths. The result is a prefix-free code for which the Kraft inequality is satisfied. Hence, Kraft inequality is satisfied for the suffix-free code as well.

Problem 5

(a) The code will be



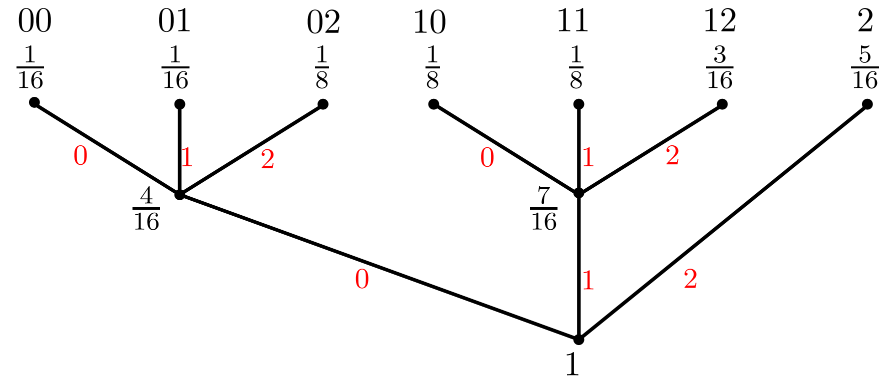
$$\begin{aligned}\overline{L} &= \sum_{i=1}^{5} p_i l_i \\ &= \frac{2}{15}\times 3 + \frac{2}{15}\times 3 + \frac{1}{3}\times 3 + \frac{1}{5}\times 2 + \frac{1}{5}\times 2 = \frac{34}{15} = 2.27\end{aligned} \tag{8}$$

(b)

$$\overline{L} = \sum_{i=1}^{5} p_i l_i = \frac{1}{5}(3+3+2+2+2) = 2.4 \tag{9}$$

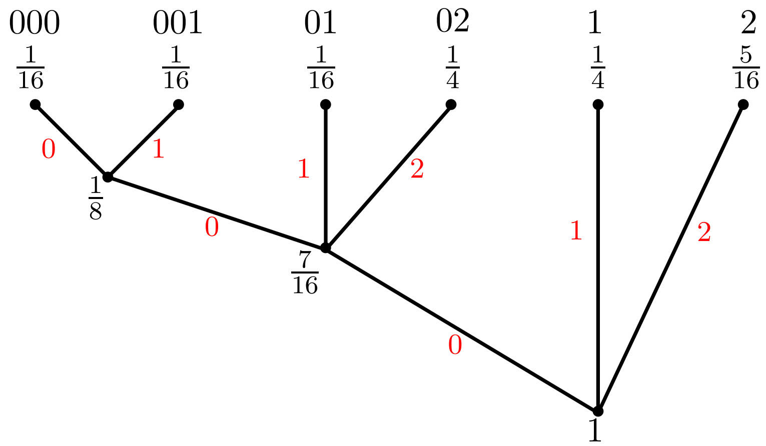
(c) (2) and (3) cannot be binary Huffman codes. note that binary Huffman codes always have their longest codewords in the form of siblings.

(d) 1. The code will be



2. Three symbols are merged at each intermediate node.

(e) 1. The code will be



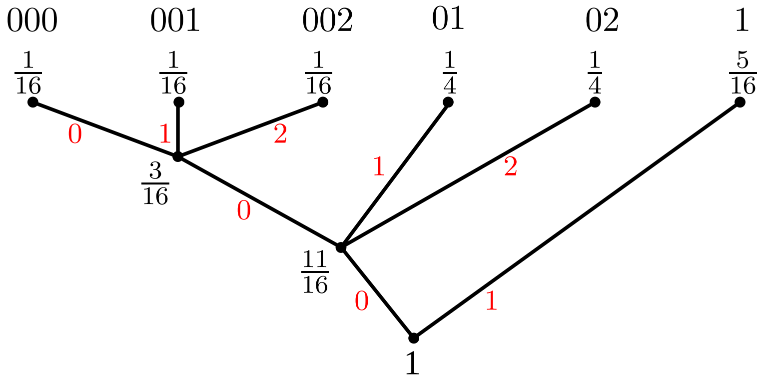
2. In the furthest intermediate node from the root of the tree, 2 symbols are merged. 3 are merged in the other steps. This is due to the fact that the number of source symbols is not of the form $1 + k(D-1)$ where $D = 3$ in our examples and thus, it is not possible to merge $D = 3$ symbols at all intermediate nodes. In part (3) we see in an example why it is better to always merge fewer than $D = 3$ symbols, if necessary, in the largest depth of the Huffman tree.

3. Consider the following two cases

- If we merge 2 symbols in depth 3 of the tree, we would have the tree drawn in part (1) and so

$$\overline{L}_1 = \sum_{i=1}^{6} p_i l_i = \frac{1}{16}\times 3 + \frac{1}{16}\times 3 + \frac{1}{16}\times 2 + \frac{1}{4}\times 2 + \frac{1}{4}\times 1 + \frac{5}{16}\times 1 = 1.56 \quad (10)$$

- If we merge 2 symbols in the root, the code would be



and

$$\overline{L}_2 = \sum_{i=1}^{6} p_i l_i = \frac{1}{16}\times 3 + \frac{1}{16}\times 3 + \frac{1}{16}\times 3 + \frac{1}{4}\times 2 + \frac{1}{4}\times 2 + \frac{5}{16}\times 1 = 1.88 \quad (11)$$

$\overline{L}_1 < \overline{L}_2$.
Note that in the first case, both the symbol with probability 1/16 and the symbol with probability 1/4 had smaller assigned codeword lengths.
This argument seems more obvious if you assume that the source produces 7 symbols instead of 6, one(which we can call "*dummy*" symbol) has probability 0. So according to Huffman procedure, it is always better to merge the "*dummy*" symbols in the first step, i.e. at the furthest intermediate node.

Problem 6

(a)

$$\Pr\{X_n = v_i | X_{n-1} = v_j, X_{n-2} = v_{i_{n-2}}, \cdots, X_0 = v_{i_0}\} = \begin{cases} \frac{1}{\deg(v_j)} & \text{if } v_i \ \& \ v_j \text{ are connected} \\ 0 & \text{else} \end{cases}$$

This is because the particle picks the edge from v_j to v_i uniformly at random among the d_j adjacent edges of v_j; and this is really independent of how the particle has reached v_j.

$$\Pr\{X_n = v_i | X_{n-1} = v_j\} = \begin{cases} \frac{1}{\deg(v_j)} & \text{if } v_i \ \& \ v_j \text{ are connected} \\ 0 & \text{else} \end{cases}$$

And this verifies $\{X_n\}$ being a Markov chain.

(b)

$$\mathbf{P} = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{3} & 0 & \frac{1}{2} \\ 0 & \frac{1}{3} & \frac{1}{3} & 0 \end{bmatrix} \quad (12)$$

(c) Call the vector of stationary distribution μ.
So we have

$$\mu\mathbf{P} = \mu \Rightarrow \mu = \begin{pmatrix} \frac{2}{10} & \frac{3}{10} & \frac{3}{10} & \frac{2}{10} \end{pmatrix}$$

(d)
- X_n is a Markov chain as argued in (a).
- $P_{v_i|v_j} = \begin{cases} \frac{1}{d_j} & \text{if } v_i \text{ \& } v_j \text{ are connected} \\ 0 & \text{else} \end{cases}$
- Take $\mu = \{\mu_j\}_{j=1}^m$, where

$$\mu_j = \frac{d_j}{\sum_{j=1}^m d_j}$$

and see that

$$\begin{aligned} \sum_{j=1}^m \mu_j P_{ji} = \mu_i &= \sum_{j=1}^m \frac{d_j}{\sum_{j=1}^m d_j} \big(\frac{1}{d_j} \mathbb{1}_{\{v_j \text{ \& } v_i \text{ connected}\}} \big) \\ &= \sum_{j=1}^m \frac{1}{\sum_{j=1}^m d_j} \mathbb{1}_{\{v_j \text{ \& } v_i \text{ connected}\}} \\ &= \mu_i \end{aligned} \tag{13}$$