BASICS:

Given: a probability distribution on a finite alphabet

(Ex.: \{ "A", "B", "C" \}, \[ P_A = \frac{1}{4}, P_B = \frac{1}{3}, P_C = \frac{5}{12} \])

Want: a variable length code, i.e.

A correspondence alphabet \rightarrow bit sequences such that the sequence corresponding to one symbol is never a beginning of a sequence corresponding to another symbol.

Ex.:

\[ \begin{align*}
A & \rightarrow 0 \\
\theta & \rightarrow 0 \ 1 \\
\theta & \rightarrow 1 \\
B & \rightarrow 1 0
\end{align*} \]

performance ("average length"):

\[ A.L. = \sum_{\text{symbols}} P(\text{symbol}) \cdot \text{length (symbol encoding)} \]

(Ex.: \[ 1 \cdot \frac{1}{4} + 2 \cdot \frac{1}{3} + 2 \cdot \frac{5}{12} = \frac{7}{4} \] for the code above)

useful results:

\[ \begin{align*}
& \times A.L. \geq \sum_{\text{symbols}} P(\text{symbol}) \cdot \log_2 \left( \frac{1}{P(\text{symbol})} \right) \\
& \text{"information entropy" / Shannon's entropy} \]

\[ A.L. \leq 1 + \sum_s P(s) \log_2 \left( \frac{1}{P(s)} \right) \]

\( \text{minimal A.L. is delivered by Huffman codes} \)

\[ \times \text{for the alphabet of } n \text{-length words } \]

\[ w = (s_1, s_2, ..., s_n) \text{, with } P(s_1 s_2 ... s_n) = P(s_1)P(s_2) ... P(s_n) \]

the entropy is \( H = \sum_s (\text{original entropy}) \)

i.e. the entropy is optimal average number of bits (\( n \to \infty \))
Ex. 1

You are told that two independent tosses of a fair die (sides 1, 2, 3, 4, 5, 6) have yielded a sum of 4.
How much info is this?

3 possibilities left out of 36
(1+3, 2+2, 3+1)

INFO = \log_2 \left( \frac{36}{3} \right) \approx 3.58

(If the sum = 7, INFO = \log_2 \left( \frac{36}{6} \right) \approx \frac{3.58}{3.58-1} = 2.58)

Ex. 2

Unfair coin: (P(Tail) = 0.02)

Need to encode the outcome of 100 tosses, to minimize A.L. Minimal A.L. = ?
The alphabet has 2^{100} symbols!

Entropy = \sum [0.02 \cdot \log_2 \frac{1}{0.02} + 0.98 \cdot \log_2 \frac{1}{0.98}] \approx 14.144

Hence:

14.1 < Minimal A.L < 15.2

Ex. 3

Huffman code for (\(P_1, P_2, P_3, P_4, P_5\)) = (0.1, 0.1, 0.1, 0.3, 0.4)

Assign 0/1 arbitrarily at each branching,

Ex. 4

\[ \begin{align*}
&\text{THIS IS A} \\
&\text{HUFFMAN CODE TREE.} \\
&\text{WHAT CAN BE SAID ABOUT} \\
&P_1, P_2, P_3, P_4? \\
&\left\{ \begin{align*}
&P_1 \leq P_2 \\
&P_1 \leq P_4 \\
&P_3 \leq P_2 \\
&P_3 \leq P_4 \\
\end{align*} \right. \\
&\text{BECAUSE } P_1, P_3 \text{ ARE THE TWO LONGEST} \\
&P_1 + P_3 \leq P_2 \quad \text{BECAUSE } (P_1, P_3) \text{ IS JOINED WITH } P_4 \\
&P_4 \leq P_2 \\
\end{align*} \]

\[ \begin{align*}
\text{IS THIS CONSISTENT WITH } &P_2 = 0.3? \\
\text{NO, BECAUSE THEN } &1 = P_1 + P_2 + P_3 + P_4 \leq 3P_2 = 0.9! \\
\text{IS THIS CONSISTENT WITH } &P_2 = \frac{1}{3}? \\
\text{YES, HERE'S AN EXAMPLE: } &P_2 = \frac{1}{3}, P_4 = \frac{1}{6}, P_1 = \frac{1}{6}, P_3 = \frac{1}{6} \\
\end{align*} \]

Ex. 5

\[ \begin{align*}
&\text{IP}(S_1) = \frac{1}{2} \\
&\text{IP}(S_2) = \text{IP}(S_3) = \ldots = \text{IP}(S_{100}) = \frac{1}{198} \\
&\text{WHAT IS THE LENGTH OF ENCODING } S_1 \text{ IN THE HUFFMAN CODE?} \\
&\text{ANSWER: 1} \\
\end{align*} \]

Ex. 6

\[ \begin{align*}
P_1 &= 0.24, P_2 = 0.25, P_3 = 0.28, P_4 = 0.23 \\
&\text{HOW MANY OPTIMAL LENGTH ENCODINGS?} \\
&\text{ANSWER: } 4! = 24 \]