Problem 1. Let the alphabet be $\mathcal{X} = \{a, b\}$. Consider the infinite sequence $X_1^\infty = abababababababab......$

(a) What is the compressibility of $\rho(X_1^\infty)$ using finite-state machines (FSM) as defined in class? Justify your answer.

(b) Design a specific FSM, call it M, with at most 4 states and as low a $\rho_M(X_1^\infty)$ as possible. What compressibility do you get?

(c) Using only the result in point (a) but no specific calculations, what is the compressibility of $X_1^\infty$ under the Lempel-Ziv algorithm, i.e., what is $\rho_{LZ}(X_1^\infty)$?

(d) Re-derive your result from point (c) but this time by means of an explicit computation.

Problem 2. From the notes on the Lempel-Ziv algorithm, we know that the maximum number of distinct words $c$ a string of length $n$ can be parsed into satisfies

$$n > c \log_K(c/K^3)$$

where $K$ is the size of the alphabet the letters of the string belong to. This inequality lower bounds $n$ in terms of $c$. We will now show that $n$ can also be upper bounded in terms of $c$.

(a) Show that, if $n \geq \frac{1}{2}m(m-1)$, then $c \geq m$.

(b) Find a sequence for which the bound in (a) is met with equality.

(c) Show now that $n < \frac{1}{2}c(c+1)$.

Problem 3. Let $U_1, U_2, \ldots$ be the letters generated by a memoryless source with alphabet $\mathcal{U}$, i.e., $U_1, U_2, \ldots$ are i.i.d. random variables taking values in the alphabet $\mathcal{U}$. Suppose the distribution $p_U$ of the letters is known to be one of the two distributions, $p_1$ or $p_2$. That is, either

(i) $\Pr(U_i = u) = p_1(u)$ for all $u \in \mathcal{U}$ and $i \geq 1$, or

(ii) $\Pr(U_i = u) = p_2(u)$ for all $u \in \mathcal{U}$ and $i \geq 1$.

Let $K = |\mathcal{U}|$ be the number of letters in the alphabet $\mathcal{U}$, let $H_1(U)$ denote the entropy of $U$ under (i), and $H_2(U)$ denote the entropy of $U$ under (ii). Let $p_{j,\text{min}} = \min_{u \in \mathcal{U}} p_j(u)$ be the probability of the least likely letter under distribution $p_j$. For a word $w = u_1 u_2 \ldots u_n$, let $p_j(w) = \prod_{i=1}^n p_j(u_i)$ be its probability under the distribution $p_j$, define $p_j(\text{empty string}) = 1$. Let $\hat{p}(w) = \max_{j=1,2} p_j(w)$.

(a) Given a positive integer $\alpha$, let $S$ be a set of $\alpha$ words $w$ with largest $\hat{p}(\cdot)$. Show that $S$ forms the intermediate nodes of a $K$-ary tree $T$ with $1 + (K - 1)\alpha$ leaves. [Hint: if $w \in S$ what can we say about its prefixes?]
Let \( W \) be the leaves of the tree \( T \), by part (a) they form a valid, prefix-free dictionary for the source. Let \( H_1(W) \) and \( H_2(W) \) be the entropy of the dictionary words under distributions \( p_1 \) and \( p_2 \).

(b) Let \( Q = \min_{v \in S} \hat{p}(v) \). Show that for any \( w \in W \), \( \hat{p}(w) \leq Q \).

(c) Show that for \( j = 1, 2 \), \( H_j(W) \geq \log(1/Q) \).

(d) Let \( W_1 \) be the set of leaves \( w \) such that \( p_1(\text{parent of } w) \geq p_2(\text{parent of } w) \). Show that \( |W_1|Qp_{1,\min} \leq 1 \).

(e) Show that \( |W| \leq \frac{1}{Q}(1/p_{1,\min} + 1/p_{2,\min}) \).

(f) Let \( E_j[\text{length}(W)] \) denote the expected length of a dictionary word under distribution \( j \). The variable-to-fixed-length code based on the dictionary constructed above emits

\[
\rho_j = \frac{\lceil \log |W| \rceil}{E_j[\text{length}(W)]} \text{ bits per source letter}
\]

if the distribution of the source is \( p_j \). Show that

\[
\rho_j < H_j(U) + \frac{1 + \log(1/p_{1,\min} + 1/p_{2,\min})}{E_j[\text{length}(W)]}.
\]

(Hint: relate \( \log |W| \) to \( H_j(W) \) and recall that \( H_j(W) = H_j(U)E_j[\text{length}(W)] \).)

(g) Show that as \( \alpha \) gets larger, this method compresses the source to its entropy for both the assumptions (i), (ii) given above.