1. **Entropy and typical sets; LZ algorithm (1 point):** Let $P$ be the probability distribution with three possible outcomes $A$, $B$, $C$ and probabilities $P(A) = 1/2$, $P(B) = 1/4$, $P(C) = 1/4$. Let $X_1, X_2$ be independent and identically distributed (IID) according to $P$.

(a) Compute $H(X_1)$, $H(X_2)$, and $H(X_1, X_2)$.
(b) Make a table that lists the joint probability $P(x_1, x_2)$ and the quantity $\frac{1}{2} \log \frac{1}{P(x_1, x_2)}$ for all possible outcomes $x_1$ and $x_2$.
(c) Compute $H_\delta(X_1, X_2)$ for $\delta = \frac{3}{8}$.
(d) Write down all elements of the typical set $T_{2,\varepsilon}(P)$ for $\varepsilon = 0.12345$.

Now for something else. Imagine running the Lempel-Ziv algorithm on the following string:

```
X00X0X0XOXXX000X⊥
```

(e) List the distinct phrases that the LZ algorithm splits the string into.
(f) List the pairs $(k, x)$ generated by the LZ algorithm.

Notation: Write $k$ as an integer in decimal notation and $x \in \{X, O, ⊥\}$.

2. **Kraft-McMillan inequality (1 point):** Let $X$ be a random variable with distribution $P$. In class, we discussed that there always exists a prefix code $C$ whose codewords have length $\ell(C(x)) = \lceil \log \frac{1}{P(x)} \rceil$. Let $C_2$ be any other uniquely decodable code. Show that, for all $k$,

$$\Pr\left( \ell(C_2(X)) \leq \ell(C(X)) - k \right) \leq \frac{1}{2^k - 1}.$$

This implies no other code can produce much shorter codewords than $C$ most of the time.

Hint: Write the probability as a sum over all possible $x$ and use the Kraft-McMillan inequality.

3. **Compression algorithms (1 point):**

In this problem, you will implement two more algorithms: a ‘universal’ compression algorithm for binary images, and the Lempel-Ziv algorithm. To get started, open the notebook at [https://colab.research.google.com/github/amsqi/iit21-homework/blob/master/03-homework.ipynb](https://colab.research.google.com/github/amsqi/iit21-homework/blob/master/03-homework.ipynb) and follow the instructions.

Please submit both the notebook and a PDF printout, or provide a link to your solution on Colab. You can achieve the maximum score if your solution produces the correct output. We will only have a closer look at your code in case of problems.

This programming problem may again be a bit more difficult than last week’s; we will grade it gently.