1. Consider a distribution $X = (0.49, 0.26, 0.12, 0.04, 0.04, 0.03, 0.02)$. Create a Huffmann code for it and compare the expected word length with $H(X)$.

2. Which of the following codes cannot be a Huffmann code for any distribution?

   a) $\{0, 10, 01\}$.

   b) $\{00, 01, 10, 110\}$.

   c) $\{01, 10\}$.

3. Consider the Huffmann code for the distribution $(\frac{1}{3}, \frac{1}{5}, \frac{1}{5}, \frac{2}{15}, \frac{2}{15})$. Show that the code is also optimum for the distribution $(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$.

4. We have n products and each of them is working/broken independently with probability $p_i > \frac{1}{2}$. Using arbitrary yes/no queries we want to find all broken products.

   a) Find an appropriate lower bound on the expected number of queries needed.

   b) Fix some optimum strategy. We are following it for our products and in the end we realized that we happened to use the longest sequence of queries that our strategy prescribes. Can you conceptually describe what the last query does? I.e. between which cases did you distinguish with a yes/no?

   c) Give an upper bound on the expected number of queries needed.

5. Source coding theorem says that we cannot losslessly compress X so that the expected word length is less than $H(X)$ and that the optimum expected word length is at most $H(X) + 1$. Is this bound tight? Given any $\varepsilon > 0$ find a distribution X so that its optimum expected word length is some-function-of-ε close to $H(X) + 1$.

6. Consider an arbitrary code $C: X \to \Sigma^*$. We know from the lecture that if C is uniquely decodable, then $L(C) \geq H(X)$. What if we remove the uniquely decodable condition? Can you give any bound?

   Didactic remark. Coming up with the task statement is harder than the solution. What is wanted from you to somehow transform C into a uniquely decodable code C' and then argue about inequalities between $L(C)$, $L(C')$ and $H(X)$.

7. In this task, we will relate sampling from a distribution with entropy.

   Compression can be viewed as "given a distribution of some objects, find a representation with the shortest expected word length". Then a dual question would be "given a distribution of some objects, how many fair coin tosses do we need sample according to the distribution?"

   As an example, try to find the optimum sampling strategy for the distribution $X = (0.5, 0.25, 0.25)$ and compare your expected number of coin tosses with $H(X)$.

   From the example, you should see that what you have created can be viewed as a tree produced by Huffmann code. If you think about it a bit more, than for any distribution you can capture the mapping between sequences of coin tosses to the objects with a "random root-leaf walk" on a rooted binary tree with the following properties:

   (i) The internal vertices of the tree have exactly two children.

   (ii) The probability of reaching a leaf at depth k is 2^{-k}.

   (iii) The expected number of sampled random bits is equivalent to the expected height of the tree because each leaf is labelled with some object and you output that object on reaching that leaf.

   The tasks start here and continue on the second page.

   a) Prove that any algorithm that samples according to a distribution X requires at least $H(X)$ random fair bits. It will be useful to think of the expected depth of the binary tree described above instead of entropy but you need to relate them first.

b) Prove that if the distribution X has the property that any probability is actually a power of two, then you actually have a matching upper bound.

c) What if the probabilities in the distribution X are not all power of twos? Somehow argue that there is a strategy that uses at most $H(X) + 2$ random bits. The idea here is not that difficult but I don't know any proof that is not based on tedious arithmetics. If you come up with a combinatorial proof, that would be great (but I cannot guarantee that it exists).