Elements of Information Theory, HUJI
Lossless Compression Programming Project

In this project you will build a lossless variable length encoder, and a corresponding decoder:

- The encoder receives a file X consisting of a sequence of bytes as input, and should convert it to a binary output file Y = Enc(X).
- The decoder receives **only** the binary output file Y and should be able to reconstruct X = Dec(Y) from it.
- You may assume the input file is of length at most 2^{20} bytes. It is required that for any input X not exceeding this length we will have Dec(Enc(X)) = X (that is, that the compression will indeed be lossless).
- The compression length of the encoder/decoder pair for input X is measured by the length of Y = Enc(X) in bytes (=length in bits/8).
- There are 4 different sample files in the project's folder in Moodle. Run your encoder on each of them, verify that the decoder reconstructs the input sequence, and measure the compression length for each of the files.
- Report to us only the compression length for each of the 4 files. We will tell you how to report the lengths soon (we will probably open a Moodle questionnaire for this).
- The teams that will obtain the best (smallest) compression lengths, will be invited to the "playoffs", where we will provide different sample files for compression, and you will compress and decompress them in front of us. The winners will be dictated according to the compression lengths in the "playoffs". We will also look at the code of the winning teams, and will want to understand how their compression algorithm works.
- Your runtime will not be measured, but compression and decompression are expected to terminate in reasonable time. As a rule of thumb, we do not want to wait more than 10 seconds for compression or decompression of a 66Kbytes file (of course in real applications the run time has to be orders of magnitude smaller than this, but here we are less concerned with computational efficiency).

Rules:

- This is a non-mandatory fun bonus project. The goal is for you to experiment with various compression techniques we have studied. Please don't take the competition too seriously.
- You may implement your encoder and decoder using any programming language you like (e.g., Python, Matlab, C, Java, whatever).
- You are not allowed to use any function that compresses. You will have to implement those functions by yourselves. "Functions that compress" include gzip, winrar, zstd etc., but also functions building Huffman trees, functions for arithmetic coding or ANS, functions for Lempel-Ziv parsing, etc. If you have doubts regarding whether or not a particular function is allowed, please ask us.
- You may do the project in groups of up to three students.
- Deadline: January 5, 2023, by 23:59. No extensions will be given.

Awards:

- 1st place: 10 points bonus to the final grade (for all team members)
- 2nd place: 8 points bonus to the final grade (for all team members)
- 3rd place: 6 points bonus to the final grade (for all team members)
- 4th place: 4 points bonus to the final grade (for all team members)
- 5th place: 2 points bonus to the final grade (for all team members)