Project 2

Information Theory 2021

Each student should write (individually) a Python 3 module `ldpc.py`, which allows the encoding and decoding of messages using LDPC codes. The module should implement three functions: `generate_code(K, eps, phi)`, `encode(K, P, w)` e `decode(K, P, y)`, with the following specification:

- The function `generate_code(K, eps, phi)` accepts a non-negative integer K, the message length, and two floats eps and phi, and should generate (using randomness) the LDPC code which was taught in class. The code is represented by a list `P` of approximate size $N \approx \frac{K}{1-\varepsilon} - K$, where each element of the list `P` is itself a list of numbers between 0 and $K-1$. The list `P[j]` contains the indices of the bits whose parity is to be bit number $K+j$ of the encoding.

- The function `encode(K,P,w)` accepts a number K, an LDPC code P, and a message w, which for the purposes of this assignment is represented as a string of `0` and `1` characters.

  The output should be another string of `0` and `1` characters, of size $K + |P|$, consisting of w concatenated with $|P|$ parities, corresponding to the encoding of message w by the code P.

- The function `decode(K, P, y)` accepts K and P as before, together with a string of `0`, `1`, and `?` characters

  The function `decode(K, P, y)` should return a string of `0` and `1` characters, of size K, which is the algorithm's best guess for which message w was encoded by P and sent via the channel. If the decoding failed, the function should return `None`. It is recommended that you use the simple decoder taught in class, but you may use a decoder based on solving the implicit system of linear equations, if you prefer.

Note the following:

- In order to generate random subsets of a certain size, you may use the function `sample` from the Python 3 module `random`. E.g. `sample(range(10), 3)` chooses 3 unique random numbers between 0 and 9.

- In order to test their code, the student should also implement a function, `transmit(x, eps)`, which returns a string equal to x but where each character is replaced with `?` with probability `eps`. This can be done by using the function `random` from the module `random`, which returns a number between `0.0` and `1.0`, and comparing that number with `eps`.