

PROJECTO 2

Teoria da Informação

Cada aluno deve escrever (individualmente) um módulo `ldpc.py` em Python 3, utilizando a biblioteca `bitstring`, para codificar e decodificar mensagens utilizando os códigos LDPC.

- O módulo terá de implementar três funções: `generate_code(K, m, v)`, `encode(K, P, w)` e `decode(K, P, y, q)`, com a seguinte especificação:

- A função `generate_code(K, m, v)` aceita três números naturais em que m divide K , e devolve um código LDPC tal como ensinado na aula 21, ou seja, devolve uma lista `P` de tamanho $v\frac{K}{m}$, em que cada elemento da lista é por sua vez uma sublista de tamanho m , de forma a que cada número entre 0 e $K - 1$ aparece em exactamente v sublistas. A lista `P` representa um código de bloco (N, K) , em que $N = K + v\frac{K}{m}$.

- A função `encode(K, P, w)` aceita um número K , um código de bloco (N, K) , em que $N = K + v\frac{K}{m}$, representado por uma lista de listas `P` gerada por uma função `generate_code(K, m, v)` (poderei usar um `P` gerado pela vossa implementação, ou um `P` gerado por mim), e um `BitArray` `w` de tamanho K .

O output deve ser um `bitarray` `x` de tamanho N , que corresponde à codificação da mensagem `w` pelo código `P`.

- A função `decode(K, P, y, q)` aceita K e `P` tais como acima, mais dois `BitArrays` `y` e `q` de tamanho N . A interpretação de `y` e `q` é a seguinte:

- * Uma palavra-código `x` de N bits foi enviada por um canal de rasura.
- * Nas posições i aonde `q[i] = 0`, temos `y[i] = x[i]`. Nas posições i aonde `q[i] = 1`, houve uma rasura (i.e. temos um ponto de interrogação nessa posição), e nessas posições rasuradas teremos sempre `y[i] = 0`, independentemente do valor de `x[i]`.

A função `decode(K, P, y, q)` deve devolver um `BitArray` de tamanho K que é a sua melhor estimativa de qual foi a mensagem `w` que foi enviada, de acordo com o algoritmo de decodificação ensinado na cadeira, e devolve `None` em caso de erro ou falha na decodificação.

Avaliação. Os critérios de avaliação serão semelhantes ao primeiro trabalho.