

UNIVERSIDADE DE ÉVORA

COMPRESSÃO E DESCOMPRESSÃO DE IMAGENS  
BINÁRIAS

## Teoria da Informação

10 de Janeiro de 2019



*Autores:*

Diogo Rafael, 37859  
Vasco Crespo, 37913

*Professor:*

Miguel Barão

# Introdução

Este é o trabalho final de Teoria de Informação que tem como objetivo fazer compressão e descompressão binária de um ficheiro .pbm (Portable Bitmap Format) sendo que a compressão vai ser feita para dentro de um ficheiro .txt utilizando a linguagem Python.

Para a **compressão**, o ficheiro .pbm é lido pelo programa e através do algoritmo LZW é então escrito num ficheiro diferente .txt tal como é feito também o cálculo da entropia, da entropia condicional e do desempenho da compressão.

Para a **descompressão**, utilizamos o ficheiro .txt previamente comprimido e, novamente através do algoritmo LZW, efetuamos a descompressão e é escrito um ficheiro .pbm.

# Algoritmo LZW

O algoritmo LZW é um algoritmo de compressão e descompressão que utiliza um dicionário com um alfabeto pré-definido, encontrando combinações desse alfabeto lendo o símbolo de uma certa posição(A, por exemplo) e próxima posição do ficheiro a comprimir(B, por exemplo).

Quando o próximo elemento já se encontra no dicionário(A), há uma inserção no dicionário do elemento atual mais o próximo elemento(A+A). Isto é executado até ao fim do ficheiro de modo a obter um ficheiro diferente, sendo este uma versão comprimida do primeiro utilizando o algoritmo. A descompressão é feita com o "inverso" deste processo, onde a partir dos índices do dicionário o ficheiro é descomprimido voltando à versão original.

Este algoritmo faz uma ompressão mais eficiente em ficheiros maiores.

|                       |   |                  |   |                              |     |    |
|-----------------------|---|------------------|---|------------------------------|-----|----|
| <b>Entrada:</b>       | a | a                | b | ab                           | aba | aa |
| <b>Saída:</b>         | 0 | 0                | 1 | 3                            | 5   | 2  |
| <b>Palavra-Código</b> |   | <b>Seqüência</b> |   | <b>Derivada da seqüência</b> |     |    |
| 0                     |   | a                |   | Inicial                      |     |    |
| 1                     |   | b                |   |                              |     |    |
| 2                     |   | aa               |   | 0 + a                        |     |    |
| 3                     |   | ab               |   | 0 + b                        |     |    |
| 4                     |   | ba               |   | 1 + a                        |     |    |
| 5                     |   | aba              |   | 3 + a                        |     |    |
| 6                     |   | abaa             |   | 5 + a                        |     |    |

## compress.py

Neste programa é lido o ficheiro pbm na função **read\_file(ficheiro)**, onde são ignoradas todas as linhas de comentário tal como as linhas que começam com "P". É adicionado a um array **new** toda a informação dessa imagem e a um array **size** o tamanho em linhas e colunas da informação contida na imagem.

Depois, é feito um novo array **compressed** onde vai ser inserida a compressão através do algoritmo LZW do ficheiro pbm-.

Finalmente, é escrita a compressão num ficheiro novo .txt (com o tamanho linha/coluna da compressão) e é calculada a entropia (condicional e normal) e o desempenho da compressão.

## decompress.py

Neste programa é lido o ficheiro txt criado com o compress.py e é feito um novo array **compressed**, desta vez com os índices do dicionário feito no compress.py e com esses índices é feita a descompressão através de LZW novamente.

# Entropia e Entropia Condicional

A entropia é uma medida de incerteza gerada por uma fonte, é calculada através da seguinte formula:

$$H(X) = E \left[ \log_2 \frac{1}{p(x)} \right] = - \sum_{x \in \mathcal{X}} p(x) \log_2 p(x).$$

\*Neste caso, o logaritmo tem base 2 porque a base é binária.

A entropia condicional é a entropia causada por uma variável sabendo a outra, sendo as duas variáveis aleatórias e é calculada através da seguinte formula:

$$H(Y|X) = - \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log_2 p(y|x)$$

## Desempenho

O desempenho é a medida que quantifica a eficácia da compressão e é calculada através da seguinte formula:

$$\frac{New - Old}{Old} * 100\%$$

\*Em que new é o tamanho novo do ficheiro após a compressão e old é o tamanho do ficheiro original

# Conclusão

De ambos os processos, tanto na compressão tal como na descompressão conseguimos averiguar que para ficheiros mais pequenos o algoritmo LZW talvez não seja o mais apropriado pois apesar da sua compressão ser realizada com sucesso, o seu desempenho não é tao eficaz como num ficheiro grande, onde se pode verificar um desempenho muito melhor.