

# UV SY26

## TP 2

Thème : Codage de Huffman

### 1 Introduction

L'objectif de ce TP est de comparer le codage de Huffman et le codage arithmétique sur différentes images. Les probabilités des pixels sont modélisées par l'histogramme de l'image calculé dans le TP précédent.

### 2 Codeur de Huffman

1. Il faut d'abord construire l'arbre à partir de la fonction :  
`[dict, avglen]=huffmandict(symboles,prob);`
2. L'encodage est ensuite réalisé par la fonction :  
`enco = huffmanenco (image, dict);`
3. Le décodage est finalement réalisé par la fonction :  
`deco = huffmandeco(enco, dict);`

A chaque image encodée, calculez le taux de compression  $1 - \frac{\text{taille finale}}{\text{taille initiale}}$  obtenu sachant qu'un pixel est codé sur 8 bits. Comment retrouver ce taux à partir de la longueur moyenne des mots codes ?

### 3 Codeur arithmétique

1. L'encodage est réalisé par la fonction :  
`code = arithenco(image, occurrences);`
2. Le décodage est finalement réalisé par la fonction :  
`decode = arithdeco(code, occurrences, nombre_de_pixels_image);`

A chaque image encodée, calculez le taux de compression  $1 - \frac{\text{taille finale}}{\text{taille initiale}}$  obtenu sachant qu'un pixel est codé sur 8 bits.

### 4 Discussion

Testez les fonctions sur les images fournies en utilisant votre fonction de calcul d'histogramme pour obtenir les probabilités. Pensez à supprimer les probabilités de valeur nulle pour améliorer le codage. Vous commenterez les taux de compression obtenus par rapport aux différentes images encodées.