

Module : Théorie de l'information**TP N°2 : Codage de Huffman et arithmétique-Compression et décompression****Objectif du TP**

Compression et décompression d'une séquence numérique, en utilisant les algorithmes de Huffman et Arithmétique. Comparaison de l'efficacité des deux algorithmes en matière de compression.

Partie 1:

Considérant l'alphabet de la source **Alpha= {1, 2, 3}** et la séquence à compresser : **Seq=3313333323**.

1. Utiliser les fonctions Matlab (correspondant à l'algorithme de compression de Huffman) :
 - o **huffmandict**
 - o **huffmanenco**

Pour compresser la séquence des symboles numériques Seq.

2. Utiliser la fonction Matlab :

- o **Huffmandeco**

Pour décompresser la séquence binaire fournie par le compresseur du Huffman.

3. Comparer la taille initiale de la séquence (1 octet pour chaque symbole) avec la taille du code fourni par le compresseur de Huffman.
4. Utiliser la fonction Matlab (correspondant à l'algorithme Arithmétique) :
 - o **arithenco** pour compresser la séquence des symboles numériques Seq.
5. Utiliser la fonction Matlab :
 - o **arithdeco** pour décompresser la séquence binaire fournie par le compresseur Arithmétique.
6. Comparer la taille initiale de la séquence (1 octet pour chaque symbole) avec la taille du code fourni par le compresseur Arithmétique.
7. Donner des constations sur l'efficacité de ces deux algorithmes en compression de la séquence.

Partie 2 : Etude et simulation du codage de Huffman

Compression et décompression d'un texte, en utilisant l'algorithme de Huffman.

Considérant l'alphabet de la source **Alpha= {espace, a, ..., z}**. Ecrire un code Matlab permettant de faire les tâches suivantes :

1. Lecture d'un fichier texte à coder (.txt) en respectant les caractères de l'alphabet considéré. La chaîne de caractères lue par l'utilisateur doit être affectée automatiquement à la variable text.
2. Calcul du vecteur correspondant aux probabilités d'apparition des caractères de l'alphabet dans le texte.
3. Affichage d'un message d'erreur au cas où la probabilité totale est différente de 1.
4. Calcul de l'information propre (quantité d'information).
5. Affichage des mots du code correspondant à chaque caractère de l'alphabet considéré (la fonction qui sera utilisée est : `huffmandict`).
6. Encodage du texte par utilisation de la fonction `huffmanenco` et sauvegarde du résultat du codage dans un fichier (.txt). Observez la taille du fichier texte originale et le fichier compressé.
7. Décodage de la séquence binaire reçue par le décodeur de Huffman en utilisant la fonction `huffmandeco`.
8. Calcul de l'entropie de la source.
9. Calcul de la longueur moyenne du code avec affichage d'un message d'erreur si la valeur d'entropie est plus grande que celle de la longueur moyenne du code.
10. Calcul de l'efficacité et la redondance du code.
11. Calcul du taux et du gain de compression.

Les symboles sont des lettres de 'a' à 'z' minuscules plus le caractère 'Espace'. Si un texte ne contient pas tous les symboles de l'alphabet, la fonction ***huffmandict*** va tenir compte des symboles qui n'apparaissent pas dans le message (ce qui implique une structure d'arbre de Huffman complexe et taille du code importante) et cela va provoquer une fausse compression puisque cette fonction ne doit recevoir que l'alphabet formé par les symboles apparaissant dans le message.