

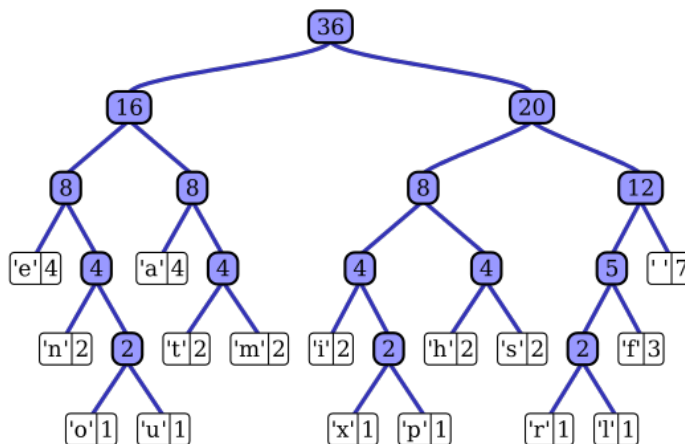
Le codage de Huffman est un algorithme de compression de données sans perte. Le codage de Huffman utilise un code à longueur variable pour représenter un symbole de la source à compresser. Par exemple pour un fichier texte, les caractères souvent présents dans le texte vont être représentés par un code de petite taille, tandis que les caractères peu présents seront représentés par un code plus long.

Le principe du codage de Huffman repose sur la création d'un arbre binaire, dont les feuilles représentent les symboles de la source (e.g. les caractères d'un fichier texte). Le chemin de la racine à une feuille dépend de la fréquence d'apparition du symbole dans la source : plus le symbole sera présent, plus le chemin sera court. Le codage d'un symbole est obtenu, en binaire, en associant le bit 0 à la branche de gauche et le bit 1 à la branche de droite.

Exemple :

- Source : « this is an example of a huffman tree »
 - Note : pour le binaire de cette source → voir encodage ASCII

- Un arbre de Huffman pour cette source :



Char ↕	Freq ↕	Code ↕
space	7	111
a	4	010
e	4	000
f	3	1101
h	2	1010
i	2	1000
m	2	0111
n	2	0010
s	2	1011
t	2	0110
l	1	11001
o	1	00110
p	1	10011
r	1	11000
u	1	00111
x	1	10010

- Code de Huffman (i.e. source compressée en binaire) :
 - 011010101000101111110001011111010001011100010010010011110011110010001110011011011111010001111011011101101110101110101110101110100010111011011000000000
 - Note : le code de l'espace est en rouge pour faciliter la lecture

Travail demandé :

1. Rapport (max 6 pages)
 - Dans votre rapport vous mettrez en avant
 1. La structure de données choisie
 2. La structuration de vos fonctions et procédures
 3. L'efficacité de votre implémentation (taux de compression cf. résultat 1 & 2) ainsi que la méthode choisie pour l'enregistrement de l'arbre
2. Résultat(s)
 - Résultat 0 : Les fonctions/procédures permettant
 1. la lecture d'un fichier texte sur le disque
 2. l'affichage du tableau des caractères, de la fréquence et du code associé
 3. la génération d'un arbre en fonction des occurrences des caractères
 - Résultat 1 : Les fonctions/procédures permettant
 1. la génération d'un arbre équilibré en terme de nombre total de caractères représentés par chaque nœud
 2. la création d'un fichier « compressé » où chaque bit est représenté par un caractère
 3. la décompression d'un fichier « compressé »
 - Résultat 2 : Les fonctions/procédures permettant
 1. la création d'un fichier compressé est en binaire
 2. la décompression d'un fichier binaire compressé

Consignes :

- Ce travail est individuel.
- Vous devez au moins atteindre le résultat 1 et rédiger le rapport.
- Il est à rendre selon le groupe, à la fin de votre dernière séance de TD. Vous rendrez sur Moodle dans un fichier .zip contenant les fichiers sources (.h et .c)
- Nom du fichier A1_GRx_NOM_PRENOM (avec x remplacé par le numéro du groupe $x \in \{A, B, C, D, E, \dots\}$ et NOM par votre nom et PRENOM par votre prénom)
- L'ensemble des codes sera analysé par un système anti-plagiat. Un plagiat entraîne la note de 0 au module.
- Un programme ne compilant pas ou ne s'exécutant pas entraîne la note de 0.