Projet : compression de données par l'algorithme de Huffman

L'objectif de ce projet à réaliser en binôme est de coder l'algorithme de Huffman (codage et décodage) présenté en cours dans le langage de programmation de votre choix. Le barème est sur 22 pts.

Echéance : le projet sera à rendre par e-mail (eric.soutil@lecnam.net) le 15 mai 2021 23 :59 au plus tard.

Travail à réaliser :

1. Récupérer les 3 fichiers `fichier1.txt` (taille 1 ko), `fichier2.txt` (taille 17 Mo), `fichier3.txt` (taille 11 Mo) à l'adresse https://cedric.cnam.fr/~soutile/SDD/Projet/
2. Demander à l'utilisateur le nom du fichier à compresser, lire le fichier à compresser et créer le tableau des fréquences de caractères. Vous devrez trier ce tableau par ordre croissant des fréquences en utilisant nécessairement le **tri par tas**. *(⇨ 5 pts)*
3. Construire l'arbre de Huffman en respectant les règles de construction indiquées plus bas et afficher les codes des caractères `'A'`, `'B'` et `'C'` (s'ils sont présents dans le fichier). *(⇨ 7 pts)*
4. Écrire le fichier compressé sur disque (extension `.huf`) et indiquer sa taille et le taux de compression (taille du fichier compressé / taille du fichier de départ). Vous devrez proposer une méthode pour stocker l'arbre de Huffman en début de fichier (ou bien les codes de chaque caractère), en vue de la décompression. Remarque : en java, pour créer un caractère (octet) correspondant par exemple au code ascii 123 (motif binaire : 1111011), on peut initialiser une variable de type char de la façon suivante : `char c = 123 ;` *(⇨ 3 pts)*
5. Lire le fichier compressé (par exemple `fichier1.txt.huf`) et le décompresser. Afin de pouvoir comparer le fichier initial au fichier décompressé, le nom du fichier décompressé devra commencer par le préfixe `dec_` (par exemple `dec_fichier1.txt`). *(⇨ 5 pts)*
6. Présenter dans un bref rapport votre travail *(⇨ 2 pts)* :
    - quelles sont les **fonctionnalités** qui ont été mises en œuvre et celles qui ne l'ont pas été parmi les points 2 à 5 précédents,
    - la description de la **structure de données** mise en œuvre pour représenter l'arbre de Huffman,
    - les **résultats** obtenus pour les 3 fichiers test,
    - la **méthode** utilisée pour stocker l'arbre ou les codes en début de fichier

Règles pour la construction de l'arbre : Les règles suivantes devront obligatoirement être respectées pour la construction de l'arbre :

- Concernant la construction de l'arbre :
    - Pour comparer deux nœuds de l'arbre :
        - On compare d'abord leurs fréquences
        - Si deux nœuds ont la même fréquence, on utilise l'ordre alphabétique des valeurs (une valeur = un caractère)
    - Lorsque le père de deux nœuds fils est créé, le père reçoit :
        - comme fils gauche (code 0) le plus petit des deux fils (au sens de la comparaison de deux nœuds de l'arbre précédemment évoquée)
        - et comme fils droit (code 1) sera nécessairement le plus grand des deux fils
        - Comme valeur (caractère) : la valeur de son fils de droite (c-à-d le plus grand)
        - Comme fréquence : la somme des fréquences de ses fils