PROJ631 – Projet algorithmique

Titre : Décompression de données codées par la méthode de Huffman

Descriptif général

Le codage de Huffman, du nom de son concepteur, est une méthode statistique de compression de données. Son principe est de remplacer un caractère (ou symbole) par une suite de bits de longueur variable. L'idée sous-jacente est de coder ce qui est fréquent sur peu de bits et au contraire ce qui est rare sur des séquences de bits plus longues. Le codage de Huffman permet une compression sans perte, c'est-à-dire qu'une suite de bits strictement identique à l'originale est restituée par décompression. Il nécessite cependant que soit connues (ou estimées) les fréquences d'apparition des différents symboles à coder. Il existe ainsi plusieurs variantes de l'algorithme de Huffman (statique, semi-adaptatif ou adaptatif) aujourd'hui utilisées dans des algorithmes de compression de fichiers tels que gzip.

Ce sujet concerne la phase de décompression de l'algorithme dans laquelle un texte compressé pourra être décodé si l'alphabet d'origine est connu ainsi que la distribution fréquentielle des caractères qui le constituent.

Descriptif détaillé

Votre programme devra réaliser la phase de décodage d'un texte compressé selon les trois étapes suivantes :

1. Lecture de l'alphabet à utiliser et des fréquences de caractères associées
2. Construction de l'arbre de codage
3. Décodage du texte compressé

puis déterminer et afficher

4. le taux de compression associé au texte initial
5. le nombre moyen de bits de stockage d'un caractère dans le texte codé

Etape 1 : Détermination de l'alphabet et des fréquences de caractères

L'alphabet est composé des caractères initialement présents dans le texte et de leur fréquence d'apparition. Le terme fréquence est ici, et dans toute la suite de l'énoncé, utilisé pour une fréquence absolue, c'est-à-dire un nombre d'occurrences des caractères de l'alphabet.

L'alphabet est fourni dans un fichier texte avec en première ligne la taille de l'alphabet (nombre de caractères) puis sur chaque ligne de fichier un caractère de l'alphabet suivi de sa fréquence. On remarquera que les caractères de l'alphabet ont été rangés par fréquence croissante puis par valeur de code ASCII (ordre alphabétique).

Etape 2 : Construction de l'arbre de Huffman

L'algorithme est décrit dans l'article de son créateur publié en 1952. Il repose sur une structure d'arbre binaire où tous les nœuds internes ont exactement deux successeurs. Les feuilles sont étiquetées avec les caractères de l'alphabet, les branches par 0 (fils gauche) et 1 (fils droit). Les chemins depuis la racine jusqu'aux feuilles constituent les codes des caractères.

La construction de l'arbre est réalisée de la manière suivante :

Créer un arbre (feuille) pour chaque caractère de l'alphabet avec la fréquence associée
Répéter
    Déterminer les 2 arbres t_1 et t_2 de fréquence minimale avec $t_1.freq \leq t_2.freq$
    Créer un nouvel arbre t avec t_1 et t_2 comme sous-arbres respectivement gauche et droite avec $t.freq = t_1.freq + t_2.freq$
Jusqu'à ce qu'il ne reste plus qu'un seul arbre

Etape 3 : Décodage du texte comprimé

Le texte comprimé est fourni dans un fichier binaire qui contient une succession d'octets qui seront tout d'accord transformés en une collection de bits à valeur 0 ou 1. La séquence de bits ainsi obtenue sera alors utilisée pour parcourir l'arbre construit à l'étape 2 de sa racine jusqu'à une feuille. Le caractère associé à la feuille atteinte est un caractère du texte initial. Le parcours d'arbre est réitéré jusqu'à la fin de la séquence de bits associée au texte compressé.

Etape 4 : Détermination du taux de compression

Le taux de compression constitue une mesure de performance de l'algorithme de Huffman relativement au texte initial. Il est défini comme le gain en volume, rapporté au volume initial des données, c'est-à-dire :

Taux de compression = Gain en volume / Volume initial = 1 – Volume final / Volume initial

Les volumes sont évalués en nombre d'octets.

Etape 5 : Détermination du nombre moyen de bits de stockage d'un caractère du texte compressé

Données fournies

L'archive fournie contient pour un texte *exemple* les deux fichiers suivants :

- *exemple*_comp.bin
- *exemple*_freq.txt

Le premier est le fichier compressé du texte *exemple* à reconstruire, le second le fichier de description de l'alphabet utilisé pour la compression.

Ces deux fichiers pourront être fournis pour d'autres textes dans le cadre d'un projet de compression. Le texte initial sera alors correctement reconstruit (compression réversible) dans la mesure où les programmes de compression et de décompression respectent les mêmes règles de construction de l'arbre de Huffman.

Résultats à fournir

Pour chacun des textes compressés fournis avec l'alphabet associé, votre programme devra générer un fichier du texte décompressé.

Article de référence

D.A. Huffman, A method for the construction of minimum-redundancy codes, Proceedings of the I.R.E., septembre 1952, pp. 1098-1102.

Sur le plagiat

Le plagiat est une forme de fraude définie dans la charte **anti-plagiat** adoptée par l'Université Savoie Mont Blanc - https://dsi.univ-smb.fr/profil/pers/charte-anti-plagiat-2014.pdf - pouvant mener à des sanctions disciplinaires. Pour lutter contre ce phénomène, l'établissement s'est doté d'un outil de détection du plagiat permettant d'évaluer le degré d'authenticité d'un document.

En particulier, dans ce module il n'est pas admissible

- de présenter un code trouvé sur internet et/ou copié d'un autre projet sans le mentionner explicitement
- de présenter un code non compris