PROJ631 – Projet algorithmique

Titre : Compression de données par codage 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. Il 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 version semi-adaptative de l'algorithme dans laquelle le texte à coder est tout d'abord lu intégralement de façon à construire l'alphabet et déterminer les fréquences d'apparition des éléments de l'alphabet.

Descriptif détaillé

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

1. Détermination de l'alphabet et des fréquences de caractères
2. Construction de l'arbre de codage
3. Codage et compression du texte initial

puis déterminer et afficher

4. le taux de compression obtenu
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 sera composé des caractères présents dans le texte fourni et uniquement de ceux-ci. La fréquence des différents caractères de l'alphabet dans le texte sera déterminée. Le terme fréquence est ici, et dans toute la suite, utilisé pour une fréquence absolue, c'est-à-dire un nombre d'occurrences des caractères dans le texte. L'ordre des caractères de l'alphabet sera maintenu par fréquence croissante puis par ordre de codage des caractères ASCII.

Etape 2 : Construction de l'arbre

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 : Codage du texte

Le code de chaque caractère est obtenu par un parcours en profondeur de l'arbre.

Chaque caractère du texte est alors codé par une succession de bits et le codage du texte est obtenu par concaténation des codes de chacun de ses caractères. Il sera stocké octet par octet dans le texte compressé.

Etape 4 : Détermination du taux de compression

Le taux de compression constitue une mesure de performance de votre algorithme relativement au texte à compresser. 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 les fichiers de texte suivants :

- textesimple.txt
- extraitalice.txt
- alice.txt

Ces fichiers correspondent à trois textes de longueur différente. Il est conseillé de mettre au point votre programme de compression en commençant par traiter un texte court (textesimple.txt) de façon à bien comprendre le mécanisme de construction de l'arbre.

Résultats à fournir

Pour chacun des textes fournis (<*nom*>.txt), votre programme devra générer un fichier du texte compressé (<*nom*>_comp.bin) et un fichier de description de l'alphabet utilisé avec les fréquences de caractère (<*nom*>_freq.txt). Ce dernier devra contenir sur une première ligne la taille de l'alphabet (nombre de caractères) puis pour chacun d'entre eux le caractère suivi de sa fréquence. Les caractères de l'alphabet seront rangés par fréquence croissante puis par valeur de code ASCII (ordre alphabétique).

Par exemple, pour le texte « bonjour!! », le fichier de fréquences serait composé des lignes suivantes :

```
7
b 1
j 1
n 1
r 1
u 1
! 2
o 2
```

Ces deux fichiers permettront dans le cadre d'un projet de décompression de retrouver le texte initial dans le mesure où les programmes de compression et de décompression respectent les mêmes règles de construction de l'arbre de Huffman.

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