Projet de programmation

Informatique pour tous, première année

Julien Reichert

Ce projet est à faire par groupes de 4 élèves maximum [1] et à rendre pour le mardi 2 mai à 8 h (GMT+2) sous forme d'une archive contenant un ou plusieurs fichiers .py et un rapport rédigé avec un vrai traitement de texte (voire mieux) expliquant en moins de quatre pages les grandes lignes du code. Il est prévu qu'au moins une séance de TP soit consacrée au projet, que ce soit pour le travail ou la soutenance.

Le sujet du projet est le codage de Huffman. Il s'agit de faire une étude comparative de l'espace nécessaire pour stocker du texte avec ou sans compression dans le cas d'un codage préfixe classique.

Le code de Huffman consiste à utiliser les fréquences d'apparition des caractères, que ce soit dans un texte ou dans l'absolu (en tenant tout de même compte de la langue), pour représenter les caractères à l'aide d'un code binaire dit préfixe (signifiant qu'aucun code d'un caractère ne soit le préfixe d'un autre). Ceci est à comparer avec un encodage naturel des caractères, par exemple ASCII (éventuellement étendu) pour lequel les codes sont tous de même taille de sorte qu'il n'y ait pas non plus d'ambigüité. Ainsi, on peut espérer avoir un texte plus court à l'aide du code de Huffman, mais il faut avoir à l'esprit que certains caractères peu fréquents auront un code plus long que leur code ASCII sauf si le nombre de caractères est suffisamment faible pour ne pas utiliser trop de codes.

Le fonctionnement du codage de Huffman est le suivant : il s'agit de construire un arbre binaire dont les feuilles sont étiquetés par les caractères à coder, et la construction se fait par fusion des arbres à disposition. Plus précisément, le poids d'un arbre est défini comme la somme des fréquences des feuilles de l'arbre, et on fusionne deux sous-arbres de poids minimal (le choix est arbitraire en cas d'égalités) en créant un nœud dont les deux fils sont les arbres à fusionner. Une fois qu'il ne reste plus qu'un arbre, on attribue à chaque caractère un code reprenant le chemin à parcourir dans l'arbre pour arriver à la feuille étiquetée par le caractère : on note dans l'ordre 0 pour chaque fois où on est descendu à gauche et 1 pour chaque fois où on est descendu à droite.

Le projet consiste à :

- Créer une structure de données adéquate pour représenter un arbre binaire.
- Écrire un algorithme de construction d'un arbre associé à un tableau de fréquences de caractères.
- Écrire un algorithme de construction du tableau de fréquences des caractères à partir d'un texte. Il faudra aussi pouvoir importer depuis un fichier un tableau de fréquences écrit sous un format au choix.
- Écrire un algorithme représentant en binaire un texte après codage et déterminer le taux de compression (suivant le tableau de fréquences employé, celui du texte ou celui de la langue) par rapport au même texte en ASCII. Variante : calculer le taux de compression par rapport au texte brut ou en comptant 5 bits par caractère et en ramenant le nombre de caractères différents utilisés à 32 (lettres de l'alphabet anglais sans majuscule, espace, virgule, trois types de points et « mise en capitale du caractère suivant », par exemple).
- Tester le tout sur un texte quelconque, mais aussi sur un extrait de *La Disparition* de Georges Perec.
- Étendre à d'autres codes préfixes (éventuellement à des codes avec séparateurs) et les comparer.

Modulariser sera une nécessité, et commenter le code (en plus du rapport) aura un impact sur la notation.

1. Le principe du travail en groupe est que les tâches soient certes réparties, mais que chacun puisse expliquer à la fin ce que les autres ont fait.