

Codage de Huffman

https://github.com/DhalEynn/PROJ631_ProjetAlgorithmique

Codage de Huffman

Le but du codage de Huffman est de réduire la taille d'un fichier texte sans perdre d'informations.

Pour cela il y a 4 grandes phases :

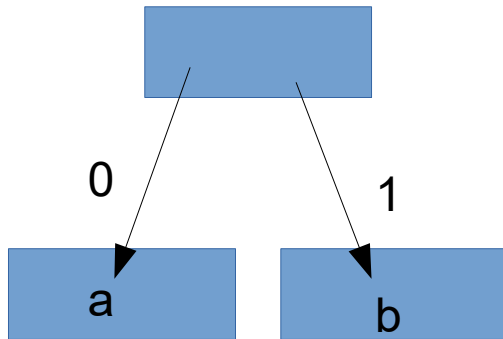
- La première consiste à récupérer l'ensemble des caractères avec leurs fréquences d'utilisation dans le texte.
- La seconde nous demande de créer un arbre contenant les différents caractères du texte. Cette arbre permet de réaliser la traduction d'un caractère vers la suite binaire le représentant.
- La troisième étape est le codage du texte en utilisant l'arbre. On remplace chaque caractère par la chaîne binaire le représentant dans l'arbre.
- Enfin, la dernière étape consiste à décoder le texte pour qu'il puisse être lu de nouveau. Pour cela on applique l'étape 2 avec un dictionnaire de fréquence préalablement transmis, puis on utilise les chaînes formées pour décoder le texte encodé.

Ainsi le codage de Huffman permet de réduire la taille d'un fichier en l'encodant, puis de le lire sans pertes en lui faisant reprendre sa taille normale.

Codage de Huffman

- Pour réaliser le codage de Huffman nous avons créé une structure nœud contenant un caractère et sa fréquence dans le texte, ainsi que ses possibles fils gauches ou droits. Si le nœud ne représente pas un caractère, la valeur du champ caractère vaut un caractère spécial '|’.
- On stocke la fréquence des différents caractères dans un fichier, que l’on utilise pour le décodage. Pour décoder, on se déplace dans l’arbre binaire un nœud par un nœud en suivant les caractères binaires du texte encodé. Dès qu’on est sur un nœud représentant un caractère alphanumérique, on écrit ce caractère dans le fichier de décodage puis on recommence au début de l’arbre avec le caractère suivant dans le fichier encodé jusqu’à avoir lu tout le fichier.

Codage de Huffman



Voici un exemple d'arbre « de Huffman » :

Le nœud de gauche contient le caractère a et le nœud de droite contient le caractère b.

L'encodage de a sera ici 0 et l'encodage de b sera 1 car a est le caractère contenu dans le fils gauche (branche 0) et b dans le fils droit (branche 1).

Le message abba s'encode donc en 0110 selon cet arbre.

abba → 0110

0110 → abba

Quand on décode, on lit le fichier encodé caractère par caractère. Ici, on lit en premier le caractère 0, donc on va sur le nœud qui contient a.

Le nœud contenant un caractère alphanumérique, on écrit la lettre a puis on recommence avec le 2ème caractère au début de l'arbre.