Mohamed Iheb faiza
Mohamed chemaou
El khamis Sadouioui

Compresseur de Huffman

Sommaire

I) Réponses aux questions

Quel est le nombre maximum de caractères (char) différents ?

Il y a 256 caractères diffèrent dont 127 affichables.

Comment représenter l'arbre de Huffman ? Si l'arbre est implémenté avec des tableaux (fg, fd, parent), quels sont les indices des feuilles ? Quelle est la taille maximale de l'arbre (nombre de noeuds) ?

L'arbre de Huffman est un tableau de structure où la structure contient les données fils droit fils gauche et parents. Les indices des feuilles vont de 0 à 255. La taille maximale de l'arbre est de 511 (256+128+64+32+16+8+4+2+1)

Comment les caractères présents sont-ils codés dans l'arbre ?

Le code de chaque caractère(une suite de et de 1) correspond à son chemin dans l'arbre

Le préfixe du fichier compressé doit-il nécessairement contenir l'arbre ou les codes des caractères ou bien les deux (critère d'efficacité) ?

Il peut contenir les deux cependant si il y a l'arbre la décompression sera moins efficace et notre fichier compresser sera plus lourds, du coup nous mettons le code des caractères dans un fichier à part

Quelle est la taille minimale de ce préfixe (expliquer chaque champ et sa longueur) ?

La taille minimale de notre préfixe est de 4 caractères « (a)1 » dans ce cas notre fichier à compresser n'a qu'un seul caractère donc le préfixe contient son code, le caractère en question est un séparateur entre les deux soit donc 4 caractères

Si le dernier caractère écrit ne finit pas sur une frontière d'octet, comment le compléter ? Comment ne pas prendre les bits de complétion pour des bits de données ?

On remplit la fin du buffer de 0 ou de 1 au choix ensuite au début de notre programme lorsqu'on récupère chaque caractère on incrémente une variable à chaque fois. Ainsi lorsqu'on décompresse à chaque caractère on décrémente notre variable et lorsqu'elle atteint zéro on ignore tous le reste du

buffer donc les bits de complétion.

Le décompresseur doit-il reconstituer l'arbre ? Comment ?

On peut mais ce n'est pas nécessaire, il est plus efficace de mettre en préfixe le tableau de codage et de s'en servir pour décoder

II) Compression

Dans un premier temps on récupère toutes nos fréquences dans un tableau.
Ensuite grâce à notre tableau de structure on génère l'arbre. Après que l'arbre ait été généré on obtient un code binaire unique de maximum 8 bits pour chaque caractère.
On crée un tableau de codage à deux dimensions où il y aura chaque caractère suivit de son code.
On crée notre fichier codes.txt qui contient les données du tableau de codage.
On crée un tableau qui sera notre buffer, on ouvre de nouveau notre fichier source et pour chaque caractère lu on écrit son code binaire dans notre buffer, a chaque fois qu'on écrit 8 bits on traduit ce code en entier, on écrit dans notre fichier compressé le caractère ASCII correspondant à cet entier et on vide notre buffer. On recommence ces étape jusqu'au dernier caractère.

III)Décompression

Pour la décompression on a décidé de mettre le préfixe à l'extérieur du fichier compressé pour avoir un gain d'espace sur ce fichier.
On ouvre notre fichier compressé on lit caractère par caractère, pour chaque caractère lu applique ces étapes :
-On traduit notre caractère ASCII en entier
-On traduit cet entier en code binaire
-On met ce code binaire dans un buffer

Une fois cette étape finit on relit notre buffer bit par bit et à chaque fois qu'on trouve un code associé à un caractère on écrit ce caractère dans notre fichier décomp.txt . Cela est possible car chaque code binaire est unique et aucun code n'est préfixe d'un autre.
Les codes se trouvent dans notre fichier code.txt qui correspond au préfixe