

Codage de Huffman

En Caml

1. Créer un fichier texte contenant un long texte en français.
2. Ouvrir un fichier avec `let fichier = open_in nomFichier`. On peut en lire un caractère à la fois avec `input_char fichier`. Que se passe-t-il lorsqu'on essaie de lire les caractères l'un après l'autre, sans condition d'arrêt ? En déduire une façon de lire le fichier entier (tester avec `print_char caractere`)
3. **Rappel** : un caractère peut être converti en entier via `int_of_char`. La conversion inverse existe également, les deux se basent sur le codage ASCII. Créer un tableau comptabilisant le nombre d'occurrences de chaque caractère. En déduire une liste de couples (caractère, occurrences).
4. Écrire une fonction de tri suivant le nombre d'occurrences sur des listes de tels couples. Trier la liste précédente.
5. On considère désormais des arbres dont les feuilles sont des caractères (sous forme des entiers les représentant). Convertir la liste précédente en liste de couples (arbre, occurrences) (toujours triée suivant le nombre d'occurrences).
6. Le codage d'Huffman consiste à utiliser un pré-traitement sur un texte représentatif du langage utilisé qui calcule un arbre dont les feuilles sont les caractères, pour pouvoir coder un texte de manière efficace. Plus le caractère est fréquent, plus sa profondeur est faible. On le construit comme suit :
 - (a) on calcule la fréquence de chaque caractère, et on obtient un ensemble de caractères avec leurs fréquences
 - (b) on trouve les deux arbres avec les fréquences f_1 et f_2 les plus petites, et on construit un arbre dont ils sont les sous-arbres. La fréquence de l'arbre associée est égale à $f_1 + f_2$.
 - (c) On répète l'opération jusqu'à obtention d'un unique arbre dont les caractères sont les feuilles

Écrire une fonction construisant cet arbre `arbreHuffman`.

7. Soit un caractère a . Il est codé par la suite de 0 et de 1 permettant de le rejoindre depuis la racine de l'arbre `arbreHuffman`, avec la convention qu'un 0 indique une recherche dans le sous-arbre gauche, un 1 une recherche dans le sous-arbre droit. Écrire deux fonctions `codage` et `decodage` permettant de passer d'un texte en français à un texte en binaire. Comparer la taille du texte d'origine (**Rappel** : coder un caractère en ASCII se fait sur 7 bits) et celle du texte codé, pour "xylophone", "tea" et pour le texte de la première question. Qu'en pensez-vous ?