

## Raffinages

**Commentaire:** Pour les listes, on utilisera un module très inspiré du module LCA\_TH dont les cellules seront composés d'une occurrence, d'un caractère, d'un sous arbre gauche et d'un sous arbre droit, on pourra donc créer des listes (sous forme de tableau) d'arbres et disposer de fonctions comme enregistrer, supprimer ou L\_Arbre (correspondant à La\_Donnee).

### Types définis :

Type T\_Octet

Type T\_Liste is tableau de T\_arbre

Type T\_Cellule is record

caractère : char;

occurence : int;

sad : T\_arbre;

sag : T\_arbre;

end record;

Type T\_arbre is pointeur T\_Cellule;

## Raffinage programme Compresser

**R0 : Compresser le fichier .txt en fichier .txt.hff**

**R1 : Comment “Compresser le fichier .txt en fichier .txt.hff” ?**

Lire le fichier et compter le nombre d'occurrence de chaque octet

Définir l'arbre de Huffman pour les caractères sur la liste

Définir la table de Huffman

Ecrire l'arbre de Huffman

Ecrire la table de Huffman

Ecrire le texte compressé à partir de son codage

Ecrire le texte compressé en octet dans un .hff

**R2 : Comment “Lire le fichier et compter le nombre d'occurrence de chaque octet” ?**

Ouvrir le fichier texte

TantQue l'on est pas arrivé à la fin du texte Faire

Caractère ← caractère parcouru du texte

Si caractère n'est pas présent dans la liste Alors

Enregistrer le caractère dans la liste

Sinon

Augmenter l'occurrence de ce caractère dans la liste

Fermer le fichier

**R2 : Comment “Définir l'arbre de Huffman pour les caractères sur la liste” ?**

Indice1, Indice 2 : Entier  
Arbre1, Arbre2 : Arbre  
Arbre\_Fusion : Arbre

Si la taille de la liste = 1 Alors  
rien

Sinon

Chercher l'indice du minimum d'occurrence Indice1 et on récupère cet Arbre1

Supprimer de la liste l'arbre1

Chercher l'indice du minimum d'occurrence Indice2 et on récupère cet Arbre2

Supprimer de la liste l'arbre2

Fusionner Arbre1 et Arbre2 en Arbre\_Fusion

Enregistrer Arbre\_fusion dans la liste

**Définir l'arbre de Huffman pour les caractères sur la liste**

## **R2: Comment "Définir la table de Huffman" ?**

Initialiser la table

Enregistrer une valeur par défaut dans la 1ere case de la table

Construire la table

Placer le caractère de fin en début de table

## **R2 : Comment "Ecrire l'arbre de Huffman" ?**

etage = 0

Si Arbre est vide Alors

rien

Sinon si etage = 0

Ecrire(" (")

Ecrire(occurrence de l'arbre)

Ecrire(")")

Si caractère de l'arbre != 0 Alors

Ecrire(" ")

Si Caractère de l'arbre = -1 Alors

Ecrire("\\$")

Sinon si caractère = 10 Alors

Ecrire("\n")

Sinon

Ecrire(valeur du caractère)

Ecrire(" ") et sauter une ligne

Sinon

Sauter une ligne

Ecrire l'arbre de huffman du sous arbre gauche

Ecrire l'arbre de huffman du sous-arbre droit

Sinon

Ecrire(" ")

Pour i allant de 1 à etage-1 Faire

Ecrire("| ")

Ecrire l'arbre en fonction du sous arbre droit ou gauche

Ecrire l'arbre en fonction des occurrence des caractères

Ecrire l'arbre de Huffman du sous arbre gauche

Ecrire l'arbre de Huffman du sous arbre droit

## **R2 : Comment "Ecrire le texte compressé à partir de son codage" ?**

Virage : Entier

Curseur : Arbre

Si Arbre est vide Alors

rien

Sinon

Compter le nombre de virages à gauche nécessaires pour arriver à la première feuille

Ajouter "0" au code à chaque fois

TantQue Virage > 0 Faire

Ajouter "1" à la signature

Décrémenter le nombre de virage

Se replacer dans l'arbre avant le dernier virage gauche

Tourner à droite

Recommencer l'écriture par récursivité avec le Curseur

Coder les caractères écrits

## **R2 : Comment "Ecrire le texte compressé en octet dans un .hff"?**

Créer un fichier .hff

Pour i allant de 1 à longueur\_texte\_comprese/8 - 1 Faire

Convertir les 8 caractère de 8i à 8(i+1) en Bits

Ecrire dans le fichier le Bits convertit en Octet

## **Raffinage Programme Décompresser**

### **R0 : Décompresser le fichier .txt.hff en .txt**

### **R1 : Comment "Décompresser le fichier .txt.hff en .txt" ?**

Récupérer le texte obtenu en fin de compression

Réécrire l'arbre

Décoder le texte

### **R2 : Comment "Récupérer le texte obtenu en fin de compression" ?**

Parcourir le fichier

Stocker chaque caractère dans une chaîne de caractère

**R2 : Comment “Réécrire l’arbre” ?**

Nbr\_0 , Nbr\_1 = 0

Tant Que Nbr\_0 >= Nbr\_1 Faire

    Si caractère du texte = 0 Alors

        Ajouter une racine à l’arbre

    Sinon

        Ajouter une feuille à l’arbre

**R2 : Comment “Décoder le texte” ?**

Pour i allant de 1 à longueur\_texte\_comprime/8 + 1 Faire

    Convertir Bits du texte\_comprime en Octet

    Parcourir la table pour retrouver cet octet

    Ecrire le caractère correspondant