PIM : Projet

Raffinages

Raffinage du programme de compression

R0 : Compresser le fichier texte .txt.

R1 : Comment "Compresser le fichier txt"?
Lire le fichier texte.
Établir le codage de Huffman du fichier texte.
Coder le texte fourni à partir de la table de Huffman.
Regrouper les bits en groupes d'octets.
Créer le fichier compressé.
Afficher la trace des principales opérations du codage de Huffman.

R2: Comment "Lire le fichier texte"?
Fichier : **in Chaîne de caractères**
Texte : **out Chaîne de caractères longue**
Ouvrir le fichier texte.
Récupérer le texte contenu dans le fichier texte.
Fermer le fichier texte.

R2: Comment "Établir le codage de Huffman du fichier texte"?
Construire le tableau de fréquences des symboles du texte. Texte : **in Chaîne de caractères longue** Table_frequences : **out Tableau**
Construire l'arbre de Huffman. Table_frequences : **in Liste Chaînée Associative** Arbre_Huffman : **out Tableau**
Construire la table de Huffman. Arbre_Huffman : **in Liste Chaînée Associative** Table_Huffman: **out Liste Chaînée associative**

R2: Comment "Coder le texte fourni à partir de la table de Huffman"?
Texte : **in Chaîne de caractères longue**

Table_Huffman: **in Liste Chaînée associative**
Texte_Code : **out Chaîne de caractères**
Code_temporaire :**in out Chaîne de caractères**

Initialiser Texte_Code à une chaîne vide.
Pour i <-1 **Jusqu'à** longueur(Texte) **Faire**
Code_temporaire <- Table_Huffman(Obtenir_Indice(Table_Huffman, Texte(i))).Code
{Obtenir_Indice(Table: **in Tableau**,Caractère: **in caractère**) retourne l'indice dans le tableau Table du caractère Caractère}
Texte_Code <- Texte_Code & "." & Code_temporaire
{On ajoute un point après chaque octet}
FinPour
Texte_Code <- Texte_Code & '.' & Table_Huffman(Obtenir_Indice(Table_Huffman, '\$')).Code

R2: Comment "Regrouper les bits en groupe d'octets"?
Texte_Code : **in Chaîne de caractères**
Texte_Compresse : **out Chaîne de caractères**

Pour i <-1 **Jusqu'à** longueur(Texte_Code) **Faire**
Ajouter les bits en groupes de 8 au texte compressé.
FinPour

R2 : Comment "Créer le fichier compressé**"**?
Ouvrir un nouveau fichier .hff.
Ajouter la liste des symboles utilisés en code ASCII.
Ajouter la structure de l'arbre de Huffman.
Ajouter le texte compressé.
Fermer le fichier .hff.

R2: Comment "Afficher la trace des principales opérations du codage de Huffman"?
Option : **in Caractère**
Si non Option = 's' **Alors**
Afficher l'arbre de Huffman.
Afficher la table de Huffman.
FinSi

R3: Comment "Construire le tableau de fréquences des symboles du texte"?
Texte : **in Chaîne de caractères longue**
Table_frequences : **out Liste Chaînée associative**
Initialiser la table de fréquences.
Pour i <-1 **Jusqu'à** longueur(Texte) **Faire**
Enregistrer la fréquence du caractère.
FinPour

Enregistrer (Table_frequences, '\$', 0)

R3: Comment "Construire l'arbre de Huffman"?

Table_frequences : **in Liste Chaînée associative**
Nombre_noeuds : **in out Entier**
Arbre_Huffman **: out Liste Chaînée associative**

Créer une liste de nœuds.
Nombre_noeuds <- Taille(Table_frequences)
Répéter
Trouver les deux nœuds avec les fréquences les plus faibles.
Combiner les nœuds en un arbre de fréquence la somme des deux fréquences.
Insérer le nœud parent dans la liste et supprimer les deux nœuds précédents.
Nombre_Noeuds <- Nombre_Noeuds - 1
Jusqu'à Nombre_Noeuds = 1

R3: Comment "Construire la table de Huffman"?

Table_Huffman **: out Liste Chaînée associative**
Arbre_Huffman **: in Liste Chaînée associative**

Parcourir en profondeur l'arbre de Huffman.
Associer à chaque feuille de l'arbre son code correspondant.
Ajouter chaque couple (symbole, code) dans Table_Huffman.

R3: Comment "Ajouter les bits en groupes de 8 au texte compressé"?

Compteur: **in out Entier**
Compteur <- 0
Si non (Texte_Code(i) = '.') **Alors**
Compteur <- Compteur +1
Texte_Compresse <- Texte_Compresse & Texte_Code(i)
Séparer les bits en octets.
FinSi

R3: Comment "Ajouter la liste des symboles utilisés en code ASCII"?

Liste_Codes : **out Tableau**
Liste_Symboles : **Tableau**
Arbre_Huffman : **in Liste Chaînée associative**

Construire la liste des symboles.
Construire la liste des codes.

R3: Comment "Ajouter la structure de l'arbre de Huffman"?

Arbre_Huffman : **in Liste Chaînée associative**
Structure : **out Chaîne de caractères**

Parcourir en profondeur l'arbre de Huffman.

R3: Comment ”Ajouter le texte compressé”?

Texte_Compresse: **in Chaîne de caractères**

Insérer le texte compressé dans le fichier .hff.

R4: Comment “Enregistrer la fréquence du caractère ”?

Si non Cle_Presente(Table_freqences,Texte(i)) **Alors**

Enregistrer(Table_frequences,Texte(i),1)

Sinon

Enregistrer(Table_frequences, Texte(i), La_Valeur(Table_frequences, Texte(i))+1)

FinSi

R4: Comment “Séparer les bits en octets”?

Si Compteur modulo 8 = 0 **Alors**

Texte_Compresse <- Texte_Compresse & ‘.’

FinSi

R4: Comment ”Construire la liste des symboles”?

Ajouter les symboles des feuilles de l’arbre de Huffman de gauche à droite(Parcours infixe).

R4: Comment “Construire la liste des codes**”**?

Liste_Codes : **out Tableau**

Liste_Symboles : **in Tableau**

Modifier chaque élément de la liste en le remplaçant par son code ASCII.

{Sauf pour le caractère ‘\$’ qui est remplacé par -1}

Remplacer -1 par son indice dans la liste et le déplacer au début de la liste.

{Le compte des indice commence à 0}

Marquer la fin de la liste en répétant le dernier élément.

Raffinage du programme de décompression

R0 : Décompresser le fichier texte .hff.

R1 : Comment "Décompresser le fichier texte .hff"

Lire le fichier .hff.
Séparer le contenu du fichier .hff.
Reconstruire la liste des caractères du texte original.
Reconstruire l'arbre de Huffman.
Restituer le texte original à partir du texte compressé.
Créer le fichier décompressé.

R2: Comment "Lire le fichier .hff"?

Fichier_Compresse : **in Chaîne de caractères**
Contenu : **out Chaîne de caractères longue**

Ouvrir le fichier .hff.
Récupérer son contenu.
Fermer le fichier .hff.

R2: Comment "Séparer le contenu du fichier .hff"?

Contenu : **in Chaîne de caractères longue**
Liste_codes **: out Tableau**
Structure_Arbre_Huffman : **out Chaîne de caractères longue**
Texte_compresse : **out Chaîne de caractères longue**
Indice : **Entier**

```
Indice <- 1
Initialiser Liste_codes.
Initialiser Structure_Arbre_Huffman.
Initialiser Texte_compresse.
TantQue non (Contenu(Indice) = '\n') Faire
    Ajouter les codes au tableau Liste_codes.
    Indice <- Indice +1
FinTQ
Indice <- Indice + 1
TantQue non (Contenu(Indice) = '\n') Faire
    Structure_Arbre_Huffman <- Structure_Arbre_Huffman &
Contenu(Indice)
    Indice <-Indice +1
FinTQ
Indice <- Indice + 1
TantQue non (Contenu(Indice) = '\n') Faire
    Texte_compresse <- Texte_compresse & Contenu(Indice)
    Indice <-Indice +1
```

FinTQ

R2: Comment ”Reconstruire la liste des caractères du texte original”?

Liste_Symboles : **out Tableau**

Liste_Codes : **in Tableau**

Initialiser Liste_Symboles.

{Capacite de Liste_Symboles = Capacité de Liste_Codes -1}

Liste_Symboles(Liste_Codes(1)+1) <- ’\$’

Pour i<-2 **jusqu’à** Liste_Codes(1)+1 **Faire**

Liste_Symboles(i-1) <- chr(Liste_Codes(i))

{chr(Liste_Codes(i)) retourne le caractère correspondant au code ASCII Liste_Codes(i)}

FinPour

Pour i<-Liste_Codes(1)+2 **jusqu’à** Capacite-1 **Faire**

{Capacite : Capacité de Liste_Codes}

Liste_Symboles(i)<-chr(Liste_Codes(i))

FinPour

R2: Comment “Reconstruire l’arbre de Huffman**“?**

Structure_Arbre_Huffman : **in Chaîne de caractères longue**

Arbre_Huffman **: out Liste chaînée associative**

Liste_Symboles **: in Tableau**

Initialiser un arbre vide.

Parcourir les données décrivant l’arbre.

Enregistrer dans chaque feuille de l’arbre le symbole correspondant de Liste_Symboles.

R2: Comment “Restituer le texte original à partir du texte compressé”?

Reconstruire la table de Huffman.

Décoder le texte compressé.

R2: Comment “Créer le fichier décompressé”?

Texte_decompresse : **in Chaîne de caractères**

Ouvrir un nouveau fichier .txt.

Ajouter le texte décompressé.

Fermer le fichier .txt.

R3: Comment “Parcourir les données décrivant l’arbre”?

Descendre dans le sous-arbre gauche pour chaque 0 rencontré dans la structure.

Insérer une feuille pour chaque 1rencontré dans la structure.

R3: Comment "Enregistrer dans chaque feuille de l'arbre le symbole correspondant de Liste_Symboles"?

Enregistrer de la gauche vers la droite les symboles selon leur ordre d'apparition dans Liste_Symboles.

R3: Comment "Reconstruire la table de Huffman**"?**

Table_Huffman **: out Liste Chaînée associative**
Arbre_Huffman **: in Liste Chaînée associative**

Parcourir en profondeur l'arbre de Huffman.
Associer à chaque feuille de l'arbre son code correspondant.
Ajouter chaque couple (code, symbole) dans Table_Huffman.

R3: Comment "Décoder le texte compressé"?

Table_Huffman **: out Liste Chaînée associative**
Texte_compresse **: in Chaîne de caractères longue**
Texte_decompresse **: out Chaîne de caractères longue**
Indice : **Entier**
Code : **Chaîne de caractères longue**

Indice <- 1
Initialiser Code.
TantQue Indice <= longueur(Texte_compresse) **Faire**
Récupérer le code binaire en ignorant les points.
Indice <- Indice + 1
FinTQ

R4: Comment "Récupérer le code binaire en ignorant les points**"?**

Si non Texte_compresse(Indice) = '.' **Alors**
Code <- Code & Texte_compresse(Indice)
Concaténer les symboles dans le texte décompressé.
FinSi

R5: Comment"Concaténer les symboles dans le texte décompressé"?

Si Cle_Presente(Table_Huffman,Code)
Texte_decompresse <- Texte_decompresse &
La_Valeur(Table_Huffman, Code)
{La_Valeur(Table_Huffman,Code) retourne le caractère correspondant au code Code}
Code <- ''
{Réinitialiser Code}
FinSi

Evaluation des raffinages par l'étudiant

		Evaluation Etudiant (I/P/A/+)	**Justification / commentaire**	**Evaluation Enseignant (I/P/A)**
Forme (D-21)	Respect de la syntaxe Ri : Comment "... une action complexe ..." ? des actions combinées avec des structures de controle Rj : ...	A		
	Verbe à l'infinitif pour les actions complexes	A		
	Nom ou équivalent pour expressions complexes	A		
	Tous les Ri sont écrits contre la marge et espacés	A		
	Les flots de données sont définis	A		
	Une seule structure de contrôle par raffinage	A		
	Pas trop d'actions dans un raffinage (moins de 6)	P		
	Bonne présentation des structures de contrôle	A		
Fond (D21-D22)	Le vocabulaire est précis	P		
	Le raffinage d'une action décrit complètement cette action	A		
	Le raffinage d'une action ne décrit que cette action	P		
	Les flots de données sont cohérents	P		
	Pas de structure de contrôle déguisée	P		
	Qualité des actions complexes	P		