



## **Département Sciences du Numérique**

### **Programmation Impérative**

Projet : Codage de Huffman

Raffinages

## 1. Compresser

### R0 : Compresser un fichier txt

#### R1 : Comment “Compresser un fichier txt” ?

- Lire un fichier txt      Liste\_Octet : in
- Compter le nombre d'occurrence de chaque caractère      Liste\_Octet : in out  
LCA\_Integer\_Octet
- Construire l'arbre de Huffman      Liste\_Tree : out LCA\_Integer\_Tree;  
Liste\_Octet : in out
- Enregistrer le Codage de Huffman      Code : in out Unbounded\_String

#### R2 : Comment “Lire un fichier txt” ?

Pour i allant de 1 au nombre de caractères  
ajouter à texte le ième caractère

#### R2 : Compter le nombre d'occurrence de chaque caractère ?

- Pour i allant de 1 à Taille(texte)
  - Enregistrer Node avec lettre en donnée et id incrémenté de 1
  - Ajouter/remplacer ce Node dans la lca

#### R2 : Comment construire l'arbre de Huffman ?

Tant que taille(lca) > 1 faire

- Chercher les deux plus petites clés
- Créer un noeud dans la lca avec comme fils gauche et fils droit les deux plus petites clés (freq de fg <= freq de fd)
- Supprimer les noeuds ayant ces deux plus petites clés de la lca

Fin Tant que

#### R2: Comment “Enregistrer le Codage de Huffman”?

- Parcourir l'arbre par la gauche jusqu'à tomber sur une feuille
- Enregistrer le caractère et son codage
- Remonter au noeud à droite précédent et refaire l'opération

## 2. Décompresser

### R0: Décompresser un fichier .hff

#### R1: Comment “ Décompresser un fichier .hff”?

- Lire un fichier .hff
  - File\_name: Chaîne de caractères
- Enregistrer les octets représentant les caractères
  - Lca\_Char : out T\_LCA
- Stocker la suite du fichier dans une variable
  - Suite : Chaîne de caractères
- Enregistrer le parcours infixe
  - Lca\_parcours : out T\_LCA
- Reconstruire l'arbre de Huffman
  - Lca\_Char, Lca\_parcours : in T\_LCA
  - Created\_tree: out T\_Tree
- Enregistrer le codage de Huffman correspondant à chaque caractère
  - Lca\_Char\_Code : out T\_LCA
  - Created\_Tree: in T\_tree
- Ecrire le texte décodé dans un fichier texte
  - Lca\_Char\_Code : in T\_LCA

#### R2: Comment “Lire un fichier .hff”?

Ouvrir(File\_Name)

#### R2: Comment “Enregistrer les octets représentant les caractères”?

- Récupération de la position du caractère de fin
  - Pos\_Caractere\_fin : out T\_Octet
- Enregistrer les octets correspondants aux caractères dans une liste chaîné
  - Octet\_Courant : in out T\_Octet
  - Octet\_Precedent : in out T\_Octet
  - Nb\_Characters : in out Integer
  - Lca\_Char : out T\_LCA

#### R2: Comment “Stocker la suite du fichier dans une variable”?

Suite<- “”;

Tant Que not EOF(File) Faire

- Lire un octet
  - Octet\_Courant : out T\_Octet
- Récupérer les bits de l'octet
  - Octet\_Courant : in out T\_Octet
  - Bit : in out T\_octet
- Ajouter à suite “1” ou “0” suivant les valeur de bits
  - Suite : out chaîne de caractère

Fin Tant Que

#### R2: Comment “Enregistrer le parcours infixe”?

index <- 1

cpt <- 1 - - compteur des 1 rencontrés

LCA\_Integer\_data.initialiser(lca\_id);

```

Tant que cpt <= nb_char Faire
    Si (Suite)(index) ='1' Alors
        LCA_Integer_data.enregistrer(lca_parcours,index, 1);
        cpt <- cpt +1
    Sinon
        LCA_Integer_data.enregistrer(lca_parcours,index,0);
    FinSi
    index <- index+1;
Fin Tant que
    - Supprimer les caractères lus de la chaîne de caractères
        Suite : in out Chaîne de caractères
        Index : in Integer

```

## **R2: Comment “Reconstruire l'arbre de Huffman”?**

Utilisation de la procédure Huffman Tree

## **R2: Comment “Enregistrer le codage de Huffman correspondant à chaque caractère”?**

Utilisation de la procédure Huffman Code

## **R2: Comment écrire le texte décodé dans un fichier texte ”?**

**code : chaîne de caractères**

```

indice_lecture<- 1;
Pour i allant de 1 à Length(Suite) Faire
    code <- Suite(indice_lecture..i);
    - Chercher si code figure comme clé
end loop;

```