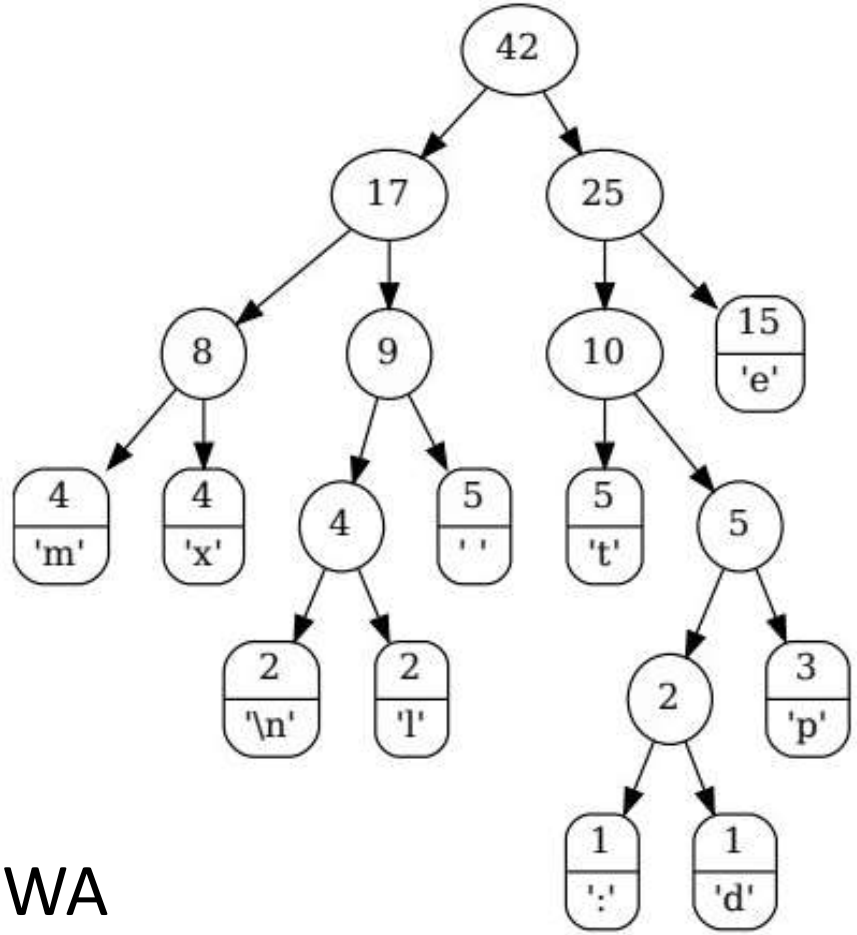
TOULOUSE INP N7

Codage Huffman
Projet Long PIM



ELLAIK FADWA

RIFAI AYA

Architecture en Modules

- Module LCA (vu en TP) pour la construction des listes de tailles dynamiques.
- Module ABR des arbres binaires (vu en TD) pour la représentation de l'arbre de Huffman.
- Emploi des bibliothèques:
  - Ada.Strings.Unbounded
  - Ada.Command_Line
  - Ada.Characters.Latin_1

Structure de Données

- TYPE T_LCA EST POINTEUR VERS T_Cellule
- TYPE T_Cellule EST UN ENREGISTREMENT avec
  Cle : T_Cle
  Valeur : T_Valeur

- TYPE T_ABR EST POINTEUR VERS T_Noeud
- TYPE T_Noeud EST UN ENREGISTREMENT avec
  Symbole : T_Symbole
  Frequence : T_Frequence
  Gauche: T_ABR
  Droit : T_ABR

Principaux Algorithmes

- Construction de l’arbre de Huffman à partir des feuilles (Caractères et leurs fréquences).
- Construction de la table de Hufman.
- Codage Huffman d’un caractère.
- Parcours en profondeur Infixe d’un arbre (DFS).

Avancement du Projet

1. Raffinages.
2. Interface des modules LCA et ABR.
3. Implementation des modules.
4. Fichiers tests.
5. Compression et Decompression avec complément des modules suivant les nécessités de l'algorithme.
6. Fichiers tests.

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.

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.

FIN