

Projet Huffman

0.1

Généré par Doxygen 1.8.11

Table des matières

1	consignes	1
2	README	3
3	Index des structures de données	5
3.1	Structures de données	5
4	Index des fichiers	7
4.1	Liste des fichiers	7
5	Documentation des structures de données	9
5.1	Référence de la structure noeud	9
5.1.1	Description détaillée	9
5.1.2	Documentation des champs	9
5.1.2.1	fd	9
5.1.2.2	frequences	9
6	Documentation des fichiers	11
6.1	Référence du fichier compresseur/include/codeBin.h	11
6.1.1	Description détaillée	11
6.1.2	Documentation des fonctions	11
6.1.2.1	CodeBin(Noeud *arbre, int racine)	11
6.2	Référence du fichier compresseur/include/constrcArbre.h	12
6.2.1	Description détaillée	12
6.2.2	Documentation des fonctions	12
6.2.2.1	ConstrcArbre(double *tab_frequence, Noeud *arbre)	12

6.3	Référence du fichier compresseur/include/defNoeud.h	13
6.3.1	Description détaillée	13
6.3.2	Documentation des définitions de type	13
6.3.2.1	Noeud	13
6.4	Référence du fichier decompresseur/include/defNoeud.h	14
6.4.1	Description détaillée	14
6.4.2	Documentation des définitions de type	14
6.4.2.1	Noeud	14
6.5	Référence du fichier compresseur/include/frequences.h	15
6.5.1	Description détaillée	15
6.5.2	Documentation des fonctions	15
6.5.2.1	CalculFrequencesCaractere(FILE *fichier)	15
6.6	Référence du fichier compresseur/include/generation.h	16
6.6.1	Description détaillée	16
6.6.2	Documentation des fonctions	16
6.6.2.1	Generation(char **index, FILE *entre, FILE *sortie, int racine, Noeud *arbre)	16
6.7	Référence du fichier decompresseur/include/decompresseur.h	17
6.7.1	Description détaillée	17
6.7.2	Documentation des fonctions	17
6.7.2.1	Decompression(FILE *Entree, FILE *Sortie)	17

Chapitre 1

consignes

A faire

- un micro rapport papier de 1 à 2 pages décrivant les choix importants effectués PAPIER ;
- Taille originelle de la source, taille compressée et gain en pourcentage (Taille originelle : 5194 ; taille compressée : 3761 ; gain : 27.6% !)
- Fichier README qui indiquera l'état d'avancement de votre projet (et par exemple les bogues résiduels)
- les listings (fichiers sources) documentés (doxygen) PAPIER
- doxygen
- cd/latex
- make
- Archive nommée par les noms du groupe
- Envoyer à meynard@lirmm.fr et pompidor@lirmm.fr
- Quel est le nombre maximum de caractères (char) différents ?
 - Le nombre maximum de caractères est 256.
- Comment représenter l'arbre de Huffman ? Si l'arbre est implémenté avec des tableaux (fg, fd, parent), quels sont les indices des feuilles ? Quelle est la taille maximale de l'arbre (nombre de noeuds) ?
 - L'arbre de huffman est représenté par une structure possédant les variables `pere`, `fg`, `fd` et `frequencies`.
 - Si l'arbre est implémenté avec des tableaux (fg, fd, parent), les indices des feuilles corresponde au code du caractère.
 - L'arbre peut avoir au maximum 256 Noeuds.
- Comment les caractères présents sont-ils codés dans l'arbre ?
 - il sont codé par leurs code ASCII.
- Le préfixe du fichier compressé doit-il nécessairement contenir l'arbre ou les codes des caractères ou bien les deux (critère d'efficacité) ?
 - Le préfixe du fichier compressé doit contenir soit l'arbre soit les codes des caractères.
 - Stocker l'arbre est plus efficace que stoker les codes des caractères.
- Quelle est la taille minimale de ce préfixe (expliquer chaque champ et sa longueur) ?
 - Le préfixe est composé de 3 octet magiques permettant d'identifié pouvant être décompressé. 1 octet représentant le nombre de bits utile du dernier octet puis l'arbre.
- Si le dernier caractère écrit ne finit pas sur une frontière d'octet, comment le compléter ? Comment ne pas prendre les bits de complétion pour des bits de données ?
 - on le complete avc des 0 inutile.
 - Lors de la décompression on vérifie si on est a la fin du fichier, si c'est le cas on ne traite que les bits utile récupéré dans l'entête
- Le décompresseur doit-il reconstituer l'arbre ? Comment ?
 - oui, a partir de l'entête.

Chapitre 2

README

Chapitre 3

Index des structures de données

3.1 Structures de données

Liste des structures de données avec une brève description :

[noeud](#)

Noeud d'un arbre binaire 9

Chapitre 4

Index des fichiers

4.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

compresseur/include/ codeBin.h	
Déclaration de la fonction qui va générer les codes binaires	11
compresseur/include/ constrcArbre.h	
Déclaration de la fonction qui va construire l'arbre	12
compresseur/include/ defNoeud.h	
Définit la structure Noeud	13
compresseur/include/ frequences.h	
Définit la fonction de génération des fréquences	15
compresseur/include/ generation.h	
Déclaration de la fonction de génération du fichier compressé	16
decompresseur/include/ decompresseur.h	17
decompresseur/include/ defNoeud.h	
Définit la structure Noeud	14

Chapitre 5

Documentation des structures de données

5.1 Référence de la structure noeud

Noeud d'un arbre binaire.

```
#include <defNoeud.h>
```

Champs de données

- unsigned int **pere**
- unsigned int **fg**
- unsigned int **fd**
- double **frequences**

5.1.1 Description détaillée

Noeud d'un arbre binaire.

Noeud est une petite structure comportant les indices de son noeud père et de ses noeuds fils (droit et gauche). Lui est également associé la fréquence d'apparation dans un fichier du caractère associé à ce noeud.

Noeud est une petite structure comportant les indices de son noeud père et de ses noeuds fils. Lui est également associé la fréquence d'apparation dans un fichier du caractère associé à ce noeud

5.1.2 Documentation des champs

5.1.2.1 unsigned int noeud : :fd

Indices des noeuds père, fils gauche et fils droit

5.1.2.2 double noeud : :frequences

Fréquence d'apparation du caractère lié au noeud

La documentation de cette structure a été générée à partir du fichier suivant :

- compresseur/include/[defNoeud.h](#)

Chapitre 6

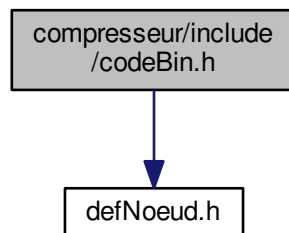
Documentation des fichiers

6.1 Référence du fichier compresseur/include/codeBin.h

Déclaration de la fonction qui va générer les codes binaires.

```
#include "defNoeud.h"
```

Graphe des dépendances par inclusion de codeBin.h :



Fonctions

— char ** [CodeBin](#) (Noeud *arbre, int racine)
Génère les codes binaires.

6.1.1 Description détaillée

Déclaration de la fonction qui va générer les codes binaires.

6.1.2 Documentation des fonctions

6.1.2.1 char** CodeBin (Noeud * *arbre*, int *racine*)

Génère les codes binaires.

Ces codes binaires sont associés à chaque caractère présent dans le fichier d'origine.

Paramètres

<i>arbre</i>	Arbre
<i>racine</i>	Indice de la racine de l'Arbre

Renvoie

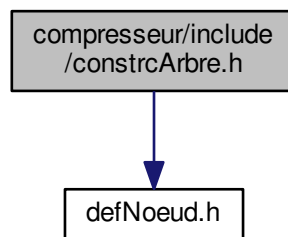
Index Tableau associant les codes binaires aux différents caractères

6.2 Référence du fichier compresseur/include/constrcArbre.h

Déclaration de la fonction qui va construire l'arbre.

```
#include "defNoeud.h"
```

Graphe des dépendances par inclusion de constrcArbre.h :



Fonctions

— int [ConstrcArbre](#) (double *tab_frequence, [Noeud](#) *arbre)
Construction de l'arbre.

6.2.1 Description détaillée

Déclaration de la fonction qui va construire l'arbre.

6.2.2 Documentation des fonctions

6.2.2.1 int ConstrcArbre (double * *tab_frequence*, Noeud * *arbre*)

Construction de l'arbre.

Paramètres

<i>tab_frequence</i>	Tableau des fréquences de répartition des caractères
<i>arbre</i>	Arbre préalablement initialisé

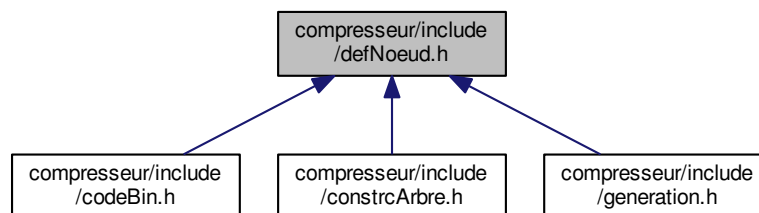
Renvoie

Ind_racine Indicé de la racine de l'arbre

6.3 Référence du fichier compresseur/include/defNoeud.h

Définit la structure Noeud.

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Structures de données

- struct `noeud`
Noeud d'un arbre binaire.

Définitions de type

- typedef struct `noeud` `Noeud`
Noeud d'un arbre binaire.

6.3.1 Description détaillée

Définit la structure Noeud.

6.3.2 Documentation des définitions de type

6.3.2.1 typedef struct `noeud` `Noeud`

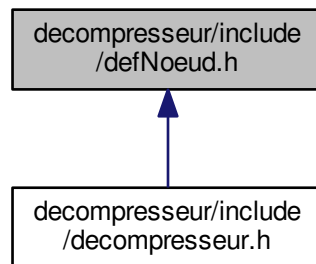
Noeud d'un arbre binaire.

Noeud est une petite structure comportant les indices de son noeud père et de ses noeuds fils (droit et gauche). Lui est également associé la fréquence d'apparation dans un fichier du caractère associé à ce noeud.

6.4 Référence du fichier decompresseur/include/defNoeud.h

Définit la structure Noeud.

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Structures de données

- struct `noeud`
Noeud d'un arbre binaire.

Définitions de type

- typedef struct `noeud Noeud`
Noeud d'un arbre binaire.

6.4.1 Description détaillée

Définit la structure Noeud.

6.4.2 Documentation des définitions de type

6.4.2.1 typedef struct noeud Noeud

Noeud d'un arbre binaire.

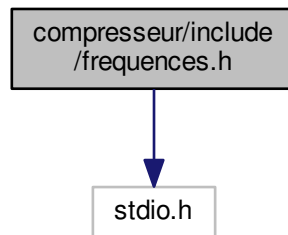
Noeud est une petite structure comportant les indices de son noeud père et de ses noeuds fils. Lui est également associé la fréquence d'apparation dans un fichier du caractère associé à ce noeud

6.5 Référence du fichier compresseur/include/frequences.h

Définit la fonction de génération des fréquences.

```
#include <stdio.h>
```

Graphe des dépendances par inclusion de frequences.h :



Fonctions

- double * [CalculFrequencesCaractere](#) (FILE *fichier)
Fonction de calcul de la répartition des fréquences dans un fichier.

6.5.1 Description détaillée

Définit la fonction de génération des fréquences.

6.5.2 Documentation des fonctions

6.5.2.1 double* CalculFrequencesCaractere (FILE * fichier)

Fonction de calcul de la répartition des fréquences dans un fichier.

Paramètres

<i>fichier</i>	Fichier où les fréquences des caractères doivent être calculées
----------------	-----------------------------------------------------------------

Renvoie

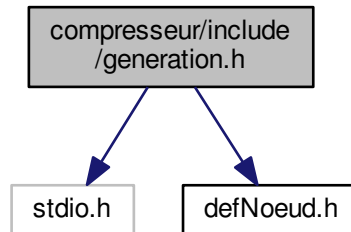
tab_frequence Un tableau contenant les fréquences des caractères contenus dans le fichier. L'indice d'une case correspond au code ASCII du caractère correspondant

6.6 Référence du fichier compresseur/include/generation.h

Déclaration de la fonction de génération du fichier compressé

```
#include <stdio.h>
#include "defNoeud.h"
```

Graphe des dépendances par inclusion de generation.h :



Fonctions

- void [Generation](#) (char **index, FILE *entre, FILE *sortie, int racine, [Noeud](#) *arbre)
Ecrit dans le fichier "compressé" les codes binaires des caractères.

6.6.1 Description détaillée

Déclaration de la fonction de génération du fichier compressé

6.6.2 Documentation des fonctions

6.6.2.1 void [Generation](#) (char ** *index*, FILE * *entre*, FILE * *sortie*, int *racine*, [Noeud](#) * *arbre*)

Ecrit dans le fichier "compressé" les codes binaires des caractères.

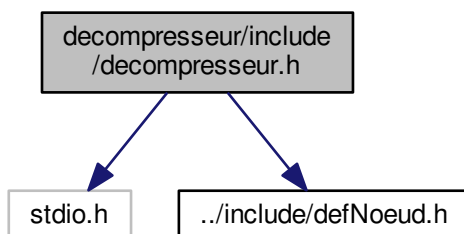
Paramètres

<i>index</i>	Tableau de codes binaires associés aux différents caractères
<i>entre</i>	Fichier d'entrée
<i>sorti</i>	Fichier de sortie (fichier compressé)
<i>racine</i>	Indice de la racine de l'arbre
<i>arbre</i>	Arbre

6.7 Référence du fichier decompresseur/include/decompresseur.h

```
#include <stdio.h>
#include "../include/defNoeud.h"
```

Graphe des dépendances par inclusion de decompresseur.h :



Macros

— #define **ERR_FORMAT** 1
Définit `ERR_FORMAT = 1`.

Fonctions

— int **Decompression** (FILE *Entree, FILE *Sortie)
Fonction permettant de décompresser un fichier.

6.7.1 Description détaillée

Déclaration de la fonction de décompression

6.7.2 Documentation des fonctions

6.7.2.1 int Decompression (FILE * Entree, FILE * Sortie)

Fonction permettant de décompresser un fichier.

Paramètres

<i>Entree</i>	Fichier compressé d'entrée
<i>Sortie</i>	Fichier de sortie décompressé

Renvoie

Retourne ERR_FORMAT si ce n'est pas un fichier qui peut être décompressé, retourne 0 sinon

Index

- CalculFrequencesCaractere
 - frequences.h, [15](#)
- CodeBin
 - codeBin.h, [11](#)
- codeBin.h
 - CodeBin, [11](#)
- compresseur/include/codeBin.h, [11](#)
- compresseur/include/constrcArbre.h, [12](#)
- compresseur/include/defNoeud.h, [13](#)
 - Noeud, [13](#)
- compresseur/include/frequences.h, [15](#)
- compresseur/include/generation.h, [16](#)
- ConstrcArbre
 - constrcArbre.h, [12](#)
- constrcArbre.h
 - ConstrcArbre, [12](#)
- decompresseur.h
 - Decompression, [17](#)
- decompresseur/include/decompresseur.h, [17](#)
- decompresseur/include/defNoeud.h, [14](#)
 - Noeud, [14](#)
- Decompression
 - decompresseur.h, [17](#)
- fd
 - noeud, [9](#)
- frequences
 - noeud, [9](#)
- frequences.h
 - CalculFrequencesCaractere, [15](#)
- Generation
 - generation.h, [16](#)
- generation.h
 - Generation, [16](#)
- Noeud
 - compresseur/include/defNoeud.h, [13](#)
 - decompresseur/include/defNoeud.h, [14](#)
- noeud, [9](#)
 - fd, [9](#)
 - frequences, [9](#)