Institut Universitaire de Technologie
Aix*Marseille Université

Site d'Arles année 2017-2018

Groupe : Escales Loïc
Marmier Baptiste

M3103 :
(Algorithmique avancée)

Projet : Codage de Huffman

Sommaire :

Introduction :

L’objectif de ce projet consistait à utiliser le codage de Huffman afin de compresser et de décompresser le contenu d’un fichier texte à l’aide d’un corpus.

Pour ce faire, nous devions écrire un programme en C++ et qu’il soit possible de l’utiliser en ligne de commande.

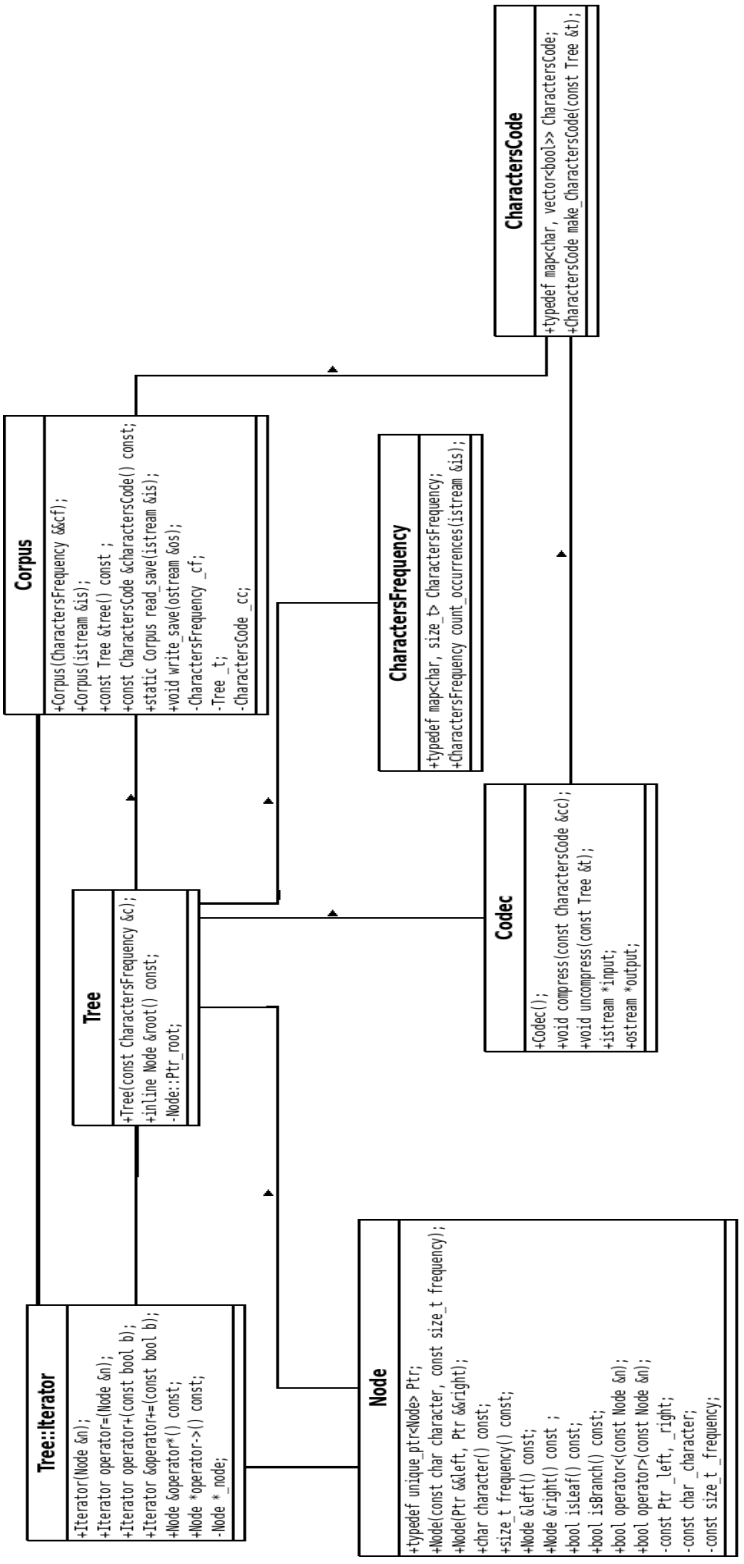
Notre programme devait accepter des paramètres avec comme minimum, au moins un paramètre pour invoquer la compression et le nom du fichier de sortie, et un paramètre pour invoquer la décompression.

En exemple le projet devait permettre :

Pour la compression → huff -c texte.txt -o texte.hff

Pour la décompression → huff -x texte.hff

Présentation des classes :



Tree

Il représente l'arbre d'Huffman et il est créé à partir de la fréquence d'apparition des caractères.

Node

C'est un nœud qui compose l'arbre d'Huffman, cela signifie qu'il possède deux fils, un caractère et la fréquence du caractère.

Codec

La classe Codec contient les différents algorithmes, pour encoder les fichiers ou les décoder en encodage de Huffman.

Corpus

La classe Corpus permet de lire et écrire un fichier « corpus », dans lequel est stocké la fréquence des caractères.

Iterator

La classe Iterator permet de naviguer dans l'arbre de node en node, en montant (de la racine jusqu'au feuille) uniquement.

CharactersFrequency

CharactersFrequency est un tableau associatif qui permet de récupérer la fréquence d'un caractère.

CharactersCode

CharactersCode est un tableau associatif qui permet de récupérer pour chaque caractère le code binaire qui lui est associé.

Les deux alias possèdent des fonctions pour les créer.

Codage de Huffman :

Qu’est-ce que Huffman et pourquoi l’utiliser dans notre projet ?

Le codage de Huffman est un algorithme de compression de données sans perte. Il utilise un code à longueur variable pour représenter un symbole de la source (par exemple un caractère dans un fichier).

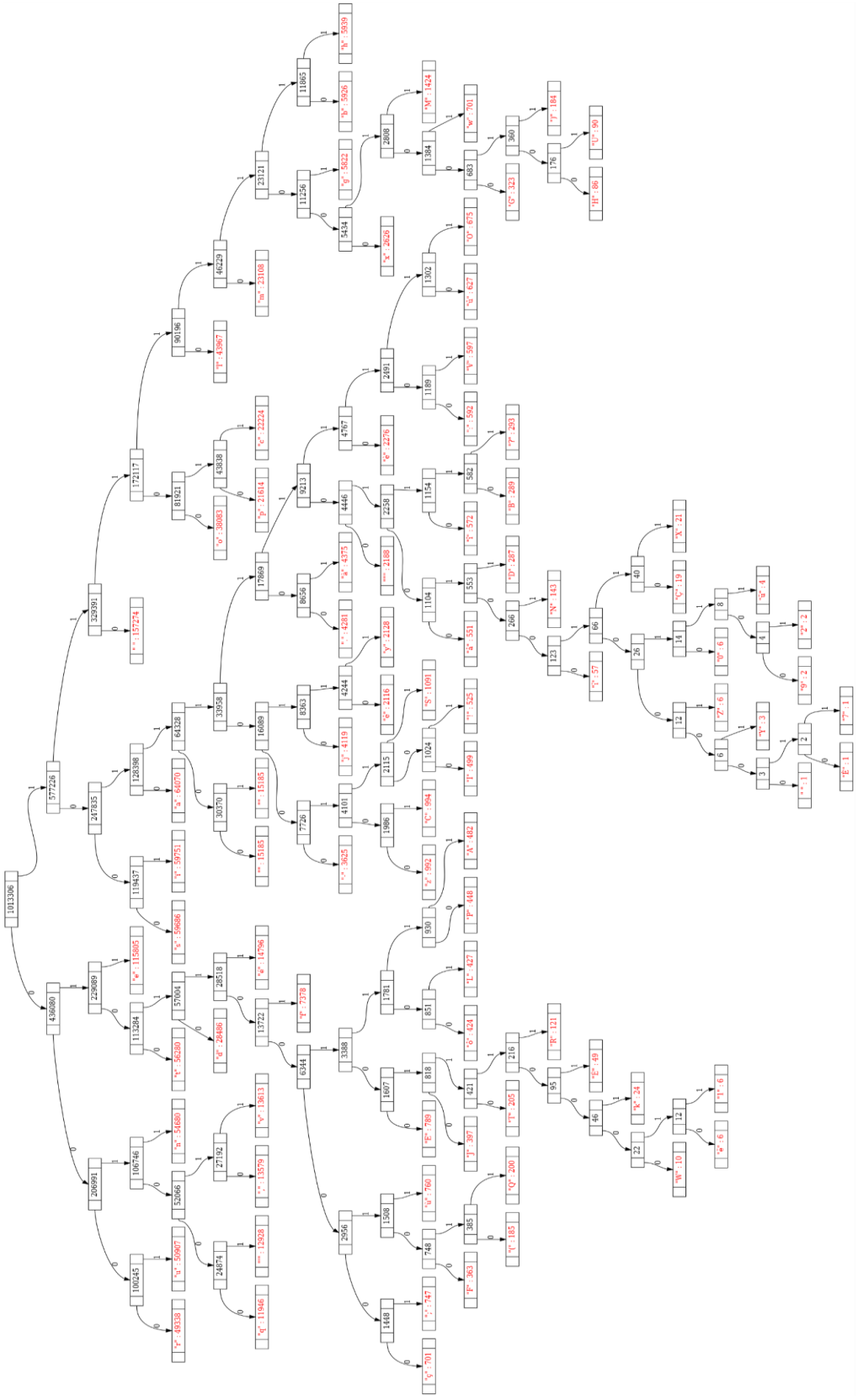
Le code est déterminé à partir d'une estimation des probabilités d'apparition des symboles de la source, plus le symbole est fréquent plus le code sera court.

Un code de Huffman est optimal au sens de la plus courte longueur pour un codage par symbole ; et une distribution de probabilité connue. Des méthodes plus complexes réalisant une modélisation probabiliste de la source permettent d'obtenir de meilleurs ratios de compression.

Le principe du codage de Huffman repose sur la création d'une structure d'arbre composée de nœuds.

Pour l’objectif de ce projet, il semblait que le codage de Huffman était la meilleure solution pour aboutir à un résultat à la fois probant et surtout fonctionnel, ainsi que dans l’optimisation du programme dans sa vitesse de recherche de compression et de décompression.

Interprétation du résultat



Conclusion :

Problèmes rencontrés

Aucun problème particulier n'a été rencontré, en effet les tps le cours ainsi que les pages de documentation sur les arbres binaires, la bibliothèque de la STL, sur le corpus, nous ont permis de mener à bien le projet.

Bilan du projet

Ce projet fut une bonne expérience, la programmation d'un arbre ainsi que le premier projet avec un délai plutôt court mais sans réel support tout près (faire les recherches par nous-mêmes et ne plus se reposer sur nos professeurs), a été un grand changement et nous a permis de voir la réalité des projets dans les entreprises (une demande, une recherche, une mise en accord, un développement).

La mise en place des classes et l'implémentation du codage d'Huffman ont été une expérience enrichissante. Le principe de recherche avec un projet court mais intense était un bon challenge.