

Projet de Programmation Impérative

Codage de Huffman

Ahmed EL MAZROUA et Abdellah BENALI

2024-2025

Résumé

Ce projet implémente l'algorithme de compression de Huffman en Ada. Il permet de compresser des fichiers en utilisant un codage à longueur variable basé sur la fréquence d'apparition des octets, et de les décompresser ensuite. Le projet met en œuvre des concepts avancés de programmation comme la gestion de la mémoire dynamique, les structures de données génériques et la manipulation de bits. Les programmes développés offrent une interface en ligne de commande avec des options de verbosité configurables.

Table des matières

1	Introduction	3
1.1	Objectifs du projet	3
1.2	Contexte	3
2	Architecture du projet	3
2.1	Vue d'ensemble	3
2.2	Description des modules	3
2.2.1	Module Arbre_Huffman	3
2.2.2	Module File_Priorite	3
3	Choix techniques	4
3.1	Structures de données	4
3.1.1	Arbre de Huffman	4
3.1.2	File de priorité	4
4	Algorithmes principaux	4
4.1	Construction de l'arbre de Huffman	4
4.2	Compression	4
4.3	Décompression	4
5	Tests et validation	4
5.1	Stratégie de test	4
5.2	Résultats des tests	5

6	Organisation du travail	5
6.1	Répartition des tâches	5
6.2	Gestion du projet	5
7	Bilan	5
7.1	Bilan technique	5
7.2	Difficultés rencontrées	5
7.3	Bilan personnel	6
7.3.1	Temps consacré	6
7.3.2	Enseignements tirés	6
8	Perspectives	6
8.1	Améliorations possibles	6
8.2	Extensions envisageables	6

1 Introduction

L'objectif principal du projet est la réalisation d'un système de compression et décompression de fichiers en utilisant l'algorithme de Huffman. Cette approche permet une compression sans perte en attribuant des codes de longueur variable aux différents symboles en fonction de leur fréquence d'apparition.

1.1 Objectifs du projet

Le projet se décompose en deux programmes principaux :

- Un compresseur qui analyse un fichier, construit un arbre de Huffman et génère un fichier compressé
- Un décompresseur qui reconstruit l'arbre à partir des métadonnées et restaure le fichier original

1.2 Contexte

La compression de données est un domaine fondamental en informatique, particulièrement pertinent dans le contexte actuel de croissance continue du volume de données numériques.

2 Architecture du projet

2.1 Vue d'ensemble

Le projet est structuré en plusieurs modules interconnectés, chacun ayant une responsabilité spécifique dans le processus de compression/décompression.

2.2 Description des modules

2.2.1 Module `Arbre_Huffman`

Ce module gère la structure centrale de l'algorithme de Huffman. Il définit :

- La structure des nœuds de l'arbre
- Les opérations de création et manipulation
- Les fonctions d'affichage pour le mode verbeux

2.2.2 Module `File_Priorite`

Module générique implémentant une file de priorité :

- Structure paramétrable par type et fonction de comparaison
- Gestion dynamique de la capacité
- Interface complète d'opérations

3 Choix techniques

3.1 Structures de données

3.1.1 Arbre de Huffman

L'arbre est implémenté comme une structure dynamique utilisant des pointeurs, permettant :

- Une construction efficace
- Une manipulation flexible
- Une gestion optimisée de la mémoire

3.1.2 File de priorité

Implémentée comme un tas binaire avec :

- Redimensionnement dynamique
- Complexité logarithmique pour les opérations
- Généricité pour la réutilisation

4 Algorithmes principaux

4.1 Construction de l'arbre de Huffman

1. Calcul des fréquences d'apparition des symboles
2. Initialisation de la file de priorité avec les feuilles
3. Fusion progressive des nœuds de plus faible fréquence
4. Construction de la table de codage

4.2 Compression

1. Lecture du fichier source
2. Application du codage selon la table de Huffman
3. Écriture optimisée des bits dans le fichier de sortie
4. Gestion du symbole de fin de fichier

4.3 Décompression

1. Lecture des métadonnées
2. Reconstruction de l'arbre
3. Décodage progressif des symboles
4. Écriture du fichier décompressé

5 Tests et validation

5.1 Stratégie de test

Les tests ont été réalisés à plusieurs niveaux :

- Tests unitaires des modules
- Tests fonctionnels des programmes
- Tests de performance et de robustesse

5.2 Résultats des tests

Les tests ont permis de valider :

- La correction des algorithmes
- La gestion correcte de la mémoire
- La robustesse face aux cas limites

6 Organisation du travail

6.1 Répartition des tâches

Ahmed	Abdellah
<ul style="list-style-type: none"> - Implémentation de la file de priorité - Tests de la file de priorité - Participation à la compression/décompression 	<ul style="list-style-type: none"> - Structure de l'arbre de Huffman - Tests de l'arbre - Participation à la compression/décompression

6.2 Gestion du projet

- Réunions régulières pour la synchronisation
- Utilisation de Git pour le versionnement
- Revue de code mutuelle

7 Bilan

7.1 Bilan technique

Les objectifs ont été atteints avec :

- Une implémentation fonctionnelle et robuste
- Une architecture modulaire et maintenable
- Une bonne gestion des ressources

7.2 Difficultés rencontrées

Les principales difficultés ont concerné :

- La manipulation bit à bit des fichiers
- La gestion de la mémoire dynamique
- La reconstruction de l'arbre lors de la décompression

7.3 Bilan personnel

7.3.1 Temps consacré

- Conception : 15 heures
- Implémentation : 25 heures
- Tests et débogage : 15 heures
- Documentation : 10 heures

7.3.2 Enseignements tirés

Le projet a permis d'acquérir :

- Une meilleure maîtrise de la programmation système
- Une expérience en gestion de projet
- Des compétences en travail d'équipe

8 Perspectives

8.1 Améliorations possibles

- Optimisation des performances
- Parallélisation des traitements
- Interface graphique

8.2 Extensions envisageables

- Support de formats de compression additionnels
- Compression adaptative
- Statistiques détaillées de compression