

Projet Python : Cryptage et décryptage

Le but principal de ce projet est de développer un système de codage et de décodage basé sur le code de Huffman.

Le codage d'Huffman a pour but de coder un texte en binaire préfixé qui consiste à coder chaque lettre par un mot sur 0, 1 (toujours le même pour une lettre). Etant donné le pourcentage d'occurrence de chaque lettre dans le texte à coder, l'algorithme de codage des lettres est le suivant :

- Initialement, chaque lettre est un arbre binaire ramené à un sommet étiqueté par la proportion d'occurrences de cette lettre dans le texte. Tant qu'il y a plus d'un arbre, réaliser les opérations suivantes :
 1. Considérer A_1 et A_2 les deux arbres dont les racines portent les plus petites étiquettes e_1 et e_2 .
 2. Construire un nouvel arbre A dont la racine r a pour fils les racines de A_1 et A_2 .
 3. La racine r est étiqueté par $e_1 + e_2$.
- Pour chaque noeud de l'arbre final, l'arête vers son fils gauche est étiquetée 0 et celle vers son fils droit 1.
- Le code associé à une lettre est le mot binaire composé des étiquettes sur les arêtes entre la racine de l'arbre final et la feuille étiquetée avec cette lettre.

Ce projet est composé de trois parties.

1 Partie 1 : Structures abstraites de données

L'objectif de cette première partie est de proposer une implémentation orientée objet de la structure et du comportement d'un arbre binaire étiqueté. Dans ce module d'arbre binaire, il s'agira de :

1. Proposer deux classes, la classe *Sommet* dont chaque objet est un sommet d'un arbre binaire et la classe *ArbreB* dont chaque objet est un arbre binaire.
2. Implémenter les opérations d'insertion et de suppression de sommets, de modification de l'étiquette d'un sommet, de recherche d'un élément dans l'arbre, ainsi que les opérations de fusion et de décomposition d'arbres binaires. Pensez à surcharger les opérateurs adéquats.
3. Proposer un programme de test démontrant les fonctionnalités d'un arbre binaire.
4. Développer une interface graphique qui permet, entre autres, d'afficher l'arbre binaire construit.

2 Partie 2 : Cryptage

L'objectif de cette partie est de prendre un texte en clair, en langue française, anglaise, ou autre, et de restituer un texte codé selon le code de Huffman. Votre application doit permettre de réaliser les opérations suivantes :

1. Calculer le pourcentage d'occurrence de chaque lettre de l'Alphabet dans le texte en entrée.
2. En utilisant le module d'arbre défini dans la première partie, construire un arbre binaire de cryptage à l'aide de l'algorithme donné (voir ci-dessus) et des statistiques réalisées dans le point 1.
3. Codifier un texte en entrée en utilisant le codage des lettres obtenu en 2. A noter que ce texte doit être écrit dans la même langue que le texte sur lequel les statistiques ont été faites en 1.
4. Enrichir l'interface graphique (Partie 1) pour permettre d'afficher le texte à coder et le texte codé correspondant.

3 Partie 3 : Décryptage

L'objectif de cette troisième et dernière partie est de prendre un texte codé selon le code de Huffman, et de restituer le texte en clair correspondant. Votre application devra permettre de :

- Vérifier si un texte a été codé selon l'arbre binaire construit dans Partie 2.
- Décrypter un texte codé selon le code de Huffman.
- Afficher le texte à décoder sur l'interface graphique et afficher le texte résultat de l'opération de décryptage.

4 Remise du projet

Ce travail est à réaliser en binôme. **Aucun projet en monome, trinôme,.. ne sera accepté.** Vous devez fournir le programme source commenté (un fichier compressé) et un mécanisme d'installation. Un compte rendu de 4 pages expliquant, entre autres, l'architecture de votre application vous est aussi demandé.

Dates à retenir :

- Remise du projet : vendredi 21 avril 2023
- Présentations : dernière séance de TD.