

L'objectif de ce projet est de réaliser le codage de Huffman. Il faudra :

- Réaliser les classes *Paire*, *FilePrio*, *Arbres*.
- Réaliser une classe de tests pour chacune de ces classes.
- Réaliser le module *Compresse* qui donne les fonctions permettant de lire un texte, de le compresser et de le décompresser.
- Réaliser une classe de test permettant de tester ce module.

Rendu et présentation intermédiaire: 5 avril Présentation des classes implémentant les structures de données et leurs classes de tests.

Date de rendu: 15 mai 24h

1 Structures de données et Classes

1. Réalisation de la classe *Paire* :

- (a) attributs `element` et `priorité` (un entier)
- (b) méthodes : les getter et setter.

2. Réalisation de la classe *FilePrio* :

- (a) attributs `fileprio` la liste des éléments rangés par priorité croissante.
- (b) méthodes : `estVide`, `teteFilePrio`, `queueFileprio`, `ajoutFilePrio`.

3. Réalisation de la classe *Noeud* :

- (a) attributs `valeur` l'étiquette du noeud.
- (b) méthodes : getter et setter.

4. Réalisation de la classe *Arbre* :

- (a) attributs `racine`, `droit`, `gauche`,
- (b) méthodes : `estVide`, `creerArbre`, `estFeuille`, getter et setter, `getValRac`.

2 Fonctions de compression

La module *compresse.py* contiendra les fonctions :

1. `lireFichier(nomfichier)`
2. `creerTabFreq(texte, nbcar)`
3. `creerFilePriorite(tabfreq, nbcar)`
4. `creerArbreCodage(fileprio)`
5. `creerTabCodes(arbrehufman, code, tabcodes)`
6. `coderTexte(texte, tabcodes)`
7. `decoderTexte(textecode, arbrehufman)`

3 Evaluation

3.1 Respect des spécifications

Votre code sera testé avec des classes de tests écrites par les enseignants et basées sur les spécifications. Si celles-ci ne sont pas respectées, les tests échoueront.

La qualité du code sera évaluée en fonction des règles de programmation des PEP demandées et des commentaires fournis. La qualité de vos tests sera également prise en compte.

3.2 Bonus

Un bonus sera appliqué pour la réalisation d'améliorations, par exemple :

- Codage du texte codé comme suite de bits et non pas de suite de caractères 0, 1.
- Réalisation d'une application qui crée un fichier formé de l'arbre et du texte codé, et qui le décode.
- Réalisation du codage adaptatif qui ne demande pas l'envoi d'un arbre de codage.

Le bonus a pour objectif de récompenser ceux qui sont intéressés par le projet et veulent aller un peu plus loin, il est déconseillé sinon.