**Compte rendu :**

**1. Classe Noeud :**

- La classe `Noeud` représente un nœud de l'arbre de Huffman. Chaque nœud contient un caractère (`char`), une fréquence (`freq`), des nœuds fils gauche et droit (`gauche` et `droite`), et un identifiant (`identifiant`).

- La méthode `\_\_lt\_\_` est implémentée pour permettre la comparaison de deux nœuds en fonction de leur fréquence.

**2. Construction de l'arbre de Huffman :**

- La fonction `construire\_arbre\_huffman` prend en entrée un texte et utilise un dictionnaire pour calculer les fréquences de chaque caractère.

- Les feuilles sont créés à partir des caractères et de leurs fréquences, puis une file de priorité est utilisée pour construire l'arbre de Huffman de manière efficace.

**3. Encodage du texte :**

- La fonction `encoder\_huffman` prend un texte en entrée, construit l'arbre de Huffman, puis utilise la fonction `construire\_codes\_huffman` pour obtenir les codes binaires associés à chaque caractère.

- L'encodage du texte est effectué en remplaçant chaque caractère par son code binaire correspondant.

**4. Décodage du texte :**

- La fonction `decoder\_huffman` prend un texte encoder et les codes associer pour décoder le texte original.

- Un dictionnaire inverse est utilisé pour retrouver le caractère correspondant à chaque code binaire.

**5. Construction des codes binaires :**

- La fonction `construire\_codes\_huffman` construit récursivement un dictionnaire de correspondance entre les caractères et leurs codes binaires en parcourant l'arbre de Huffman.