Exercice 1 On considère des codes de longueur fixe.

1. Pour $X = \{a, b, c\}$ on note $P(X) = \{\{\}, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$. Proposer un codage en binaire de l'ensemble $P(X)$ tel qu'on puisse reconstruire l'ensemble à partir de son codage.
2. Soit $X = \{a, b, c, d, e\}$. Combien y a-t-il de mots possibles de longueur n ? Combien y a-t-il de mots possibles de longueur $< n$?
3. On pose $X = \{0, 1, \ldots, 9, A, B, \ldots, F\}$ et $Y = \{0, 1\}$. Quelle doit être au minimum la longueur des mots d'un codage de longueur fixe de l'ensemble X par des mots sur Y ? Combien y a-t-il de tels codages de longueur fixe minimale ? Donner le nom d'un tel codage de longueur fixe de X sur Y bien connu. Dans ce code, quelle est l'écriture de ABC ? Quel mot est codé par 00111010 ?

Exercice 2 On considère des codes de longueur variable.

1. Considérons $X = \{a, b, c, d, e\}, Y = \{0, 1\}$ et le codage suivant :

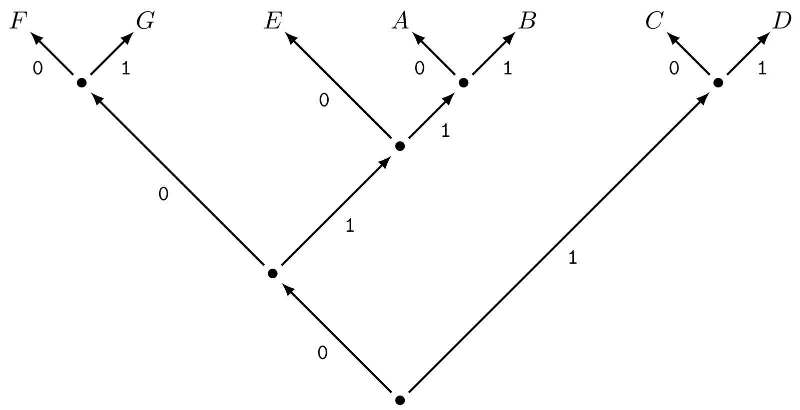
$$\tau(a) = 0 \quad \tau(b) = 10 \quad \tau(c) = 01 \quad \tau(d) = 110 \quad \tau(e) = 1011.$$

   Encoder *babc* et *bcac*. Quel est le problème ?
2. Un code préfixe est un ensemble de mots dans lequel aucun mot n'est le préfixe d'un autre.
   (a) Soit $\sigma(a) = 101000, \sigma(b) = 01, \sigma(c) = 1010$. Est-ce que σ est un code préfixe ?
   (b) Le code σ est-il suffixe ?
   (c) Tout code préfixe est uniquement déchiffrable. Est-ce que la réciproque est vraie ?
3. Pour chacun des ensembles suivants, dire s'il s'agit ou non d'un code, éventuellement préfixe ou suffixe. Justifier.
   - $E_1 = \{01, 100, 1101, 0111\}$
   - $E_2 = \{0, 10\}$
   - $E_3 = \{0, 11, 100, 101\}$
   - $E_4 = \{001, 100, 101\}$
   - $E_5 = \{01, 10, 101\}$
   - $E_6 = \{0, 011, 10\}$
   - $E_7 = \{0^n 1 : n \in \mathbb{N}\}$
   - $E_8 = \{01, 101, 110, 1110, 0100\}$
   - $E_9 = \{0, 11, 01110, 10101\}$

Exercice 3

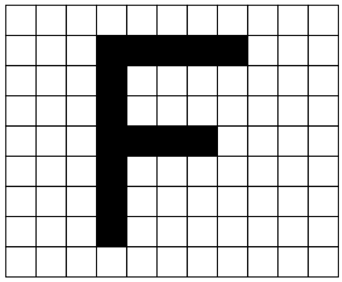
1. En utilisant l'arbre ci-dessous, décoder chacun des deux mot binaires :

$$b_1 = 1001000110010001100101001000110010001100101011011011000011011000$$
$$b_2 = 10110110110000110110000111110011100000111000011111001110000 0111000$$



2. Récupérer les fréquences de chacune des sept lettres dans le mot dont le code est $b = b_1 b_2$. Vérifier que l'arbre est bien un arbre de Huffman construit sur ces fréquences. Calculer la longueur moyenne d'un mot de code. Comparer avec un code de longueur fixe.
3. Construire deux arbres de Huffman, l'un construit à partir du mot b_1, l'autre à partir de b_2. Calculer la longueur moyenne d'un mot de code dans chacun des cas. Discuter des trois codages.

Exercice 4 On considère la représentation ci-contre, en image noir et blanc, de la lettre F. On utilise `0` pour représenter la couleur blanche et `1` pour la couleur noire. Dans la suite on suppose que les dimensions de l'image sont fixées et connues, et n'ont donc pas à être codées.
Le principe de RLE (Run-length encoding) est de remplacer chaque plage de `0` (ou de `1`) par sa longueur. On considère que l'on commence toujours par des `0`, donc l'encodage de `0000000111` est 7, 3 et celui de `110000` est 0, 2, 4.



1. Sans codage combien de bits faut-il pour représenter l'image ?
2. Utiliser RLE pour représenter l'image en lisant chaque ligne de gauche à droite, indépendamment et du haut vers le bas. Fournir la suite s des valeurs.
3. Si l'on code les entiers de la suite s par des mots binaires de longueur fixe minimale, quelle sera la longueur d'un tel mot ? Quelle sera la taille du codage de la suite s (c'est-à-dire de l'image) ? Dans notre cas, ce codage permet-il de compresser ? Si oui, quel est le taux de compression obtenu ?
4. À partir de la suite s, donner le tableau associant à chaque valeur y apparaissant, son nombre d'occurrences.
5. À partir du tableau, construire un arbre de Huffman correspondant.
6. Coder la suite s avec le code de Huffman associé à cet arbre. Quelle est la taille du codage de l'image ? Dans ce cas, le double codage RLE puis Huffman permet-il de compresser ? Si oui, quel est le taux de compression obtenu ?

Exercice 5 Montrer qu'à une liste de fréquences fixée (ou à un message donné) peut correspondre plusieurs arbres de Huffman dont les structures mêmes (quand on oublie les étiquettes) sont différentes.

Exercice 6 On considère une source qui émet continûment des `0` avec une probabilité $p_0 = \frac{1}{4}$ et des `1` avec une probabilité $p_1 = \frac{3}{4}$. On désire utiliser la méthode de Huffman pour compresser l'information reçue. Pour cela on regroupe par blocs de deux bits : `00`, `01`, `10` et `11`. On calcule la probabilité que chaque bloc apparaisse et on applique la méthode de Huffman.

1. Calculer le taux de compression β_2 associé.
2. On regroupe maintenant par blocs de 3 bits : calculer le taux de compression β_3 associé.

Exercice 7 Soit u le mot binaire $u_1 u_2 \cdots u_{100}$ avec $u_k = \begin{cases} \mathtt{1} & \text{si } k \text{ est un nombre premier} \\ \mathtt{0} & \text{sinon} \end{cases}$
(il n'est pas nécessaire d'écrire u explicitement pour répondre aux questions).

0. Énumérer les nombres premiers inférieurs à 100.

On découpe ce mot u en blocs de taille 2, de la forme $u_{2k-1}u_{2k}$ pour $1 \leq k \leq 50$.

1. Indiquer le nombre d'occurrences de chacun de ces blocs de taille 2 dans u.
2. Construire un arbre de Huffman sur les blocs de taille 2.
3. Calculer la longueur du mot u une fois compressé par la méthode de Huffman.
4. Écrire une méthode qui réponde à la question 1 pour des mots/blocs plus longs. Expérimenter.