TD 5 : Compression

Exercice 1 (*). Il est bien connu que les mots de passe sont des éléments importants de la sécurité informatique. Parmi les caractéristiques importantes qui influent sur la solidité d'un mot de passe, il y a sa longueur.

Voici quelques éléments qui permettent de mesurer l'effet du nombre de caractères d'un mot de passe.

- Sur un clavier standard, il y a 105 caractères imprimables (lettres minuscule ou majuscule, chiffres, accents, ponctuations, symboles).
- La vitesse du processeur d'un ordinateur en 2020 est de l'ordre de 3 GHz, ce qui signifie qu'il exécute environ 3 milliards d'opérations à la seconde.
- L'espérance de vie des humains en 2020 est de 80 ans environ.
- En considérant qu'un mot de passe est vérifié en une opération, saurez-vous calculer le nombre de caractères que doit comporter un mot de passe pour que la durée d'une vie humaine ne suffise pas à le découvrir par Recherche exhaustive avec un ordinateur actuel ?

Exercice 2 (*). Baby-LRE

Quelle est cette image ?

13													
3, 1, 5, 1, 3													
4, 1, 3, 1, 4													
3, 7, 3													
2, 2, 1, 3, 1, 2, 2													
1, 11, 1													
1, 1, 1, 7, 1, 1, 1													
1, 1, 1, 1, 5, 1, 1, 1, 1													
4, 2, 1, 2, 4													
13													

Exercice 3 (**). Code de Huffman

Le code de Huffman est un algorithme de compression de texte sans perte, inventé par David Albert Huffman en 1952. Il repose sur la construction d'un arbre binaire. L'objectif est de coder avec un minimum de bits les caractères qui apparaissent le plus souvent.

Pour construire l'arbre, il faut commencer par compter le nombre d'occurrences de chaque caractère dans le texte à compresser. Ensuite, chaque caractère et son nombre d'occurrences (appelé *poids*) sont stockés dans les (futures) feuilles d'un arbre binaire. L'arbre est alors construit comme suit, depuis les feuilles jusqu'à la racine :

- Associer deux nœuds de plus faibles poids en leur donnant pour parent un nouveau nœud dont le poids équivaut à la somme de leurs deux poids. Il faut aussi virer les deux plus petites lettres.
- Recommencer le processus jusqu'à n'avoir plus qu'un seul nœud.

Il y a évidemment plusieurs façons de réaliser concrètement cette construction, qui aboutissent à des arbres différents, mais qui permettent toutes d'obtenir un codage de Huffman correct.
Pour encoder un caractère, il suffit de lui associer le chemin dans l'arbre depuis la racine jusqu'à la feuille codant ce caractère, en représentant par 0 les branches gauches et par 1 les branches droites. Pour décoder une suite de bits, il faut aussi partir de la racine et suivre les 0 (vers la gauche) et les 1 (vers la droite) jusqu'à une feuille.
Le code de Huffman d'un texte est alors constitué de la concaténation des codes binaires de chacun des caractères qui le composent, accompagné de l'arbre binaire correspondant, avec les étiquettes des feuilles. Lorsque le texte à encoder est suffisamment long et comprend suffisamment de répétitions de caractères, le code de Huffman est plus court que le texte initial, et il s'agit bien d'une compression de texte.
L'arbre construit par l'algorithme de Huffman induit une propriété importante qui rend possible le décodage : comme les caractères se trouvent sur les feuilles de l'arbre, le code binaire d'un caractère n'est jamais le début du code d'un autre caractère (on parle de code *préfixe*). C'est pourquoi, bien que les longueurs des codes des caractères diffèrent, il n'est pas utile de les séparer dans la suite de bits transmise : le code d'un caractère se termine lorqu'une feuille est atteinte dans l'arbre.
Ainsi, le code de Huffman est un code *à longueur variable* : selon leur nombre d'occurrences dans le texte, les caractères ne sont pas forcément encodés par une suite de bits de même longueur. De plus, il est optimal : il a été prouvé que, de tous les algorithmes qui encodent caractère par caractère, il est celui qui possède le meilleur taux de compression. Ce code est notamment utilisé dans les algorithmes de compression `GZIP` et `DEFLATE`.

1. Calculez le nombre d'occurrences de chaque caractère du virelangue :

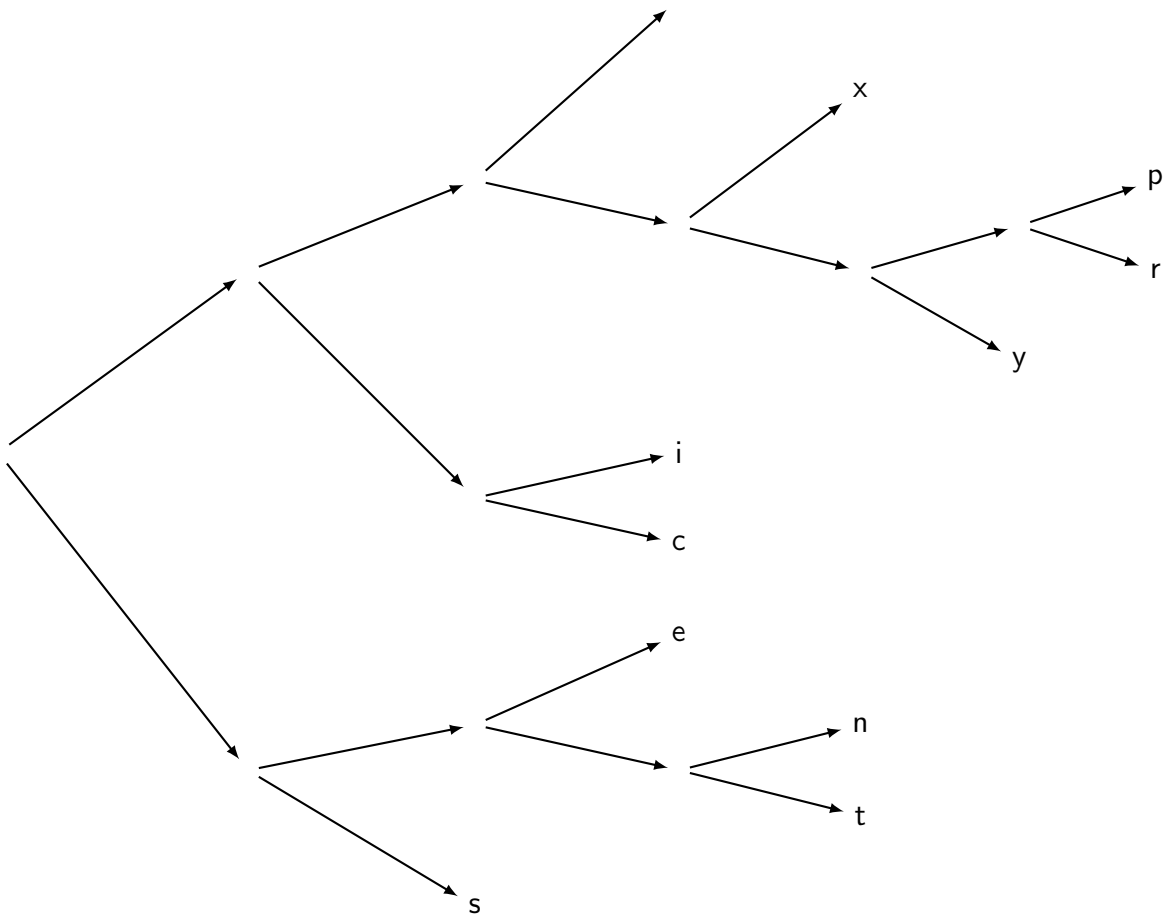
   ```
   les chaussettes de l archiduchesse sont elles seches et archiseches
   ```

   N'oubliez pas les espaces ! Combien de bits sont nécessaires pour coder cette phrase en ASCII sur 8 bits ?

2. Construisez un arbre de Huffman pour cette phrase et déduisez-en le tableau d'encodage des caractères correspondant.

3. Déterminez le code de cette phrase. Combien de bits comprend-t-il ? Comme l'arbre est spécifique à cette phrase, et qu'il est nécessaire pour le décodage, il faudrait aussi prendre en compte sa taille pour déterminer le taux exact de compression.

4. Voici un arbre de Huffman et un message compressé. Déterminez le message en français.



1101000011010001000011011010100110001101101010010101011
0001101000100000110011100110000110110011000110100010000
0111001010101100011010001000011011010100110001101101010
0101010110001101000100000111001010101100011010001000001
1001110011000011011001 1

Exercice 4 (**). Quadtree

Une technique souvent utilisée pour la compression d'images est une représentation sous forme d'arbre. Pour simplifier, les images sont supposées être en noir et blanc, carrées et de côté 2^n.

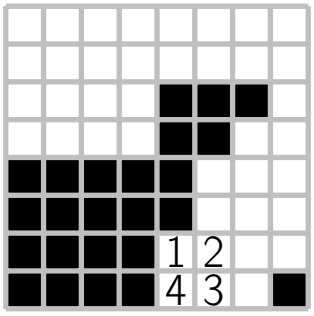
Représenter une image par un quadtree repose sur deux principes :

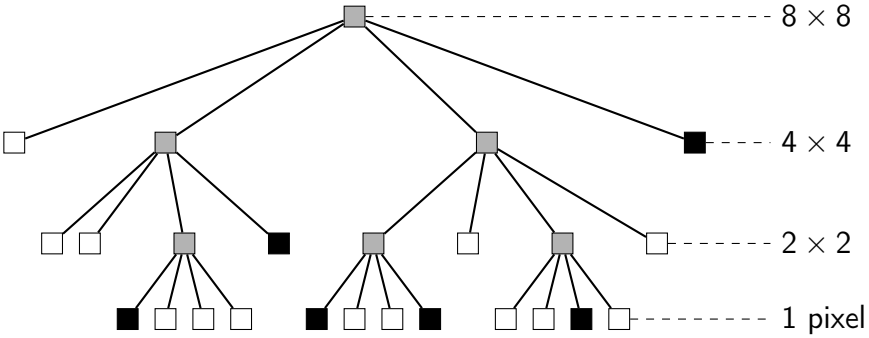
- une image toute blanche ou toute noire est représentée uniquement par sa couleur ;
- une image bicolore se divise en quatre images carrées.

L'image ci-dessous va être utilisée pour illustrer le principe de construction d'un quadtree. Les couleurs sont notées `B` pour blanc et `N` pour noir. Un nœud à quatre fils est noté `F(hg,hd,bd,bg)`, où `hg`, `hd`, `bg`, `bd` représentent respectivement la sous-image carrée en haut à gauche (1), en haut à droite (2), en bas à droite (3) et en bas à gauche (4).

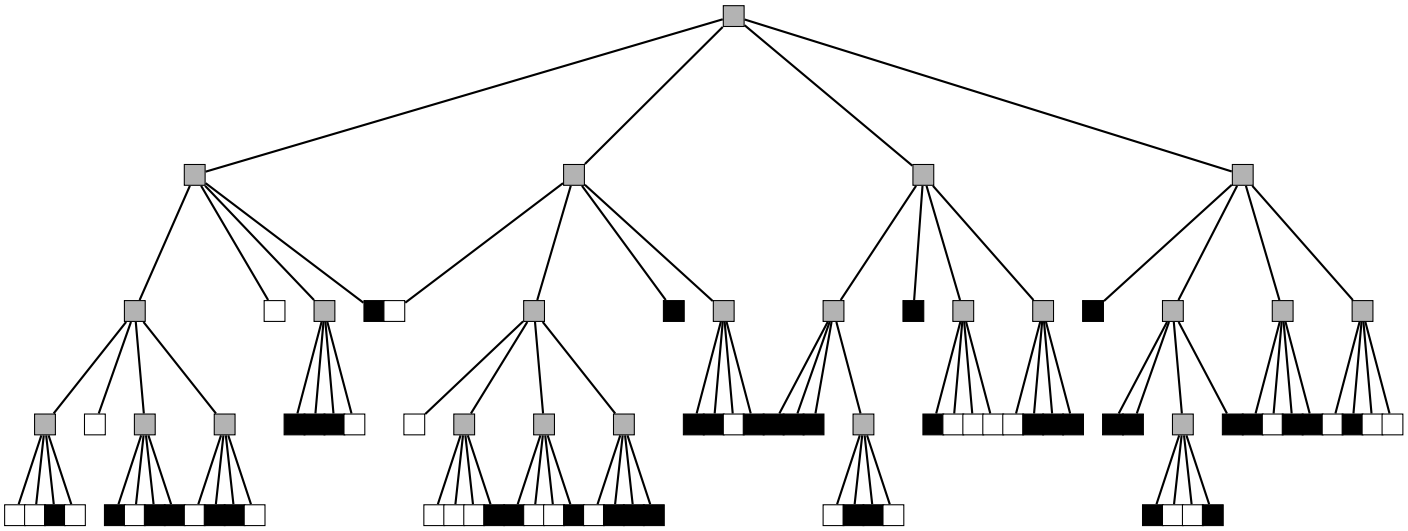
Cette image n'est ni toute noire ni toute blanche, elle est donc divisée en quatre images de tailles égales. La première image en haut à gauche est toute blanche, donc elle sera repésentée par B. La seconde image en haut à droite n'est ni toute noire ni toute blanche, donc elle est divisée en quatre images de tailles égales. La troisième image est elle aussi bicolore, donc elle sera aussi découpée en quatre images, dont la première image est une image dont la couleur des deux carrés 1 et 4 est noire. Enfin, la dernière image est toute noire et représentée par N. Ce raisonnement est appliqué aux huit images restantes afin d'obtenir l'arbre suivant dont une représentation graphique est donnée ci-dessous.

`F(B,F(B,B,F(N,B,B,B),N),F(F(N,B,B,N),B,F(B,B,N,B),B),N)`

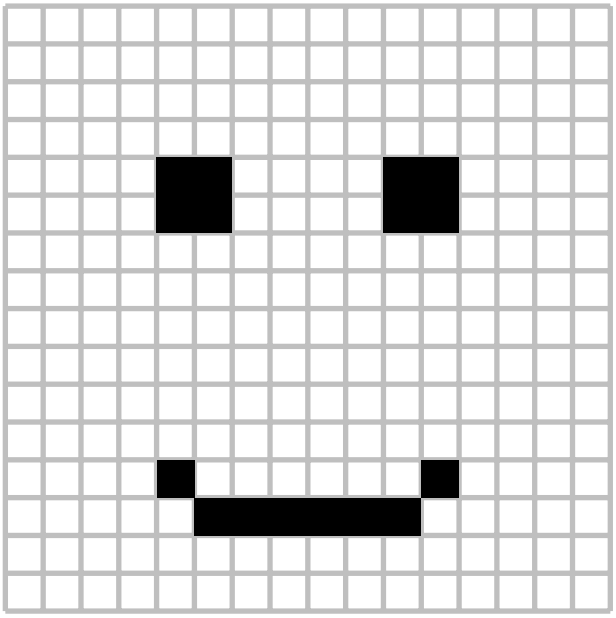




1. Quelle est l'image représentée par l'arbre ci-dessous ?



2. Quel est le quadtree associé à l'image ci-dessous ? Quelle est l'écriture récursive de l'arbre ?



Exercice 5. (***) Algorithme de Lempel-Ziv

Idée : Algorithme de compression par dictionnaire (utiliser des mots au lieu des lettres)

Exemple : "un gars bon est un bon gars" est codé par "un,gars,bon,est,1,3,2"

Il existe plusieurs variantes de cet algorithme ; LZ77 et LZ78 sont libres de droits.

Algorithme : LZ77 LZ77 utilise une fenêtre coulissante en 2 parties :

- Partie 1 : le dictionnaire
- Partie 2 : tampon de lecture.

Initialement, la fenêtre est située de façon à ce que le tampon de lecture soit sur le texte, et que le dictionnaire n'y soit pas.

À chaque itération, l'algorithme cherche dans le dictionnaire le plus long facteur qui se répète au début du tampon de lecture ; ce facteur est codé par le triplet (i, j, c) où :

- i est la distance entre le début du tampon et la position du début de la répétition dans le dictionnaire ;
- j est la longueur de la répétition ;
- c est le premier caractère du tampon différent du caractère correspondant dans le dictionnaire.

Après avoir codé cette répétition, la fenêtre coulisse de $j+1$ caractères vers la droite. Dans le cas où aucune répétition n'est trouvée dans le dictionnaire, le caractère c qui a provoqué la différence est codé alors $(0, 0, c)$.

1. Coder la séquence "abcdefabcdefabcdefabcdef" avec LZ77.
2. Coder cette séquence avec LZ77 et une taille de fenêtre de recherche tenant sur 3 bits.

Exercice 6 (*)**.** Si on expédie des bits sur un canal binaire symétrique à raison de 512 bits toutes les millisecondes, avec un taux d'erreur de 1%, quel est le nombre de bits faux reçus au bout de trois heures ?

Exercice 7. (**) Numéro INSEE

Construction : Il s'agit d'un nombre de $n = 15$ chiffres, composé comme suit : un numéro d'identification K sur $k = 13$ chiffres suivi d'une clef C de $r = 2$ chiffres, calculée de telle sorte que que $K + C$ soit un multiple de 97.

1. Quel est la clef du numéro de sécurité sociale 2.63.05.38.516.305 ?

2. Quel est le rendement de ce code ?
3. Combien d'erreurs de chiffres, la clef du numéro de sécurité sociale permet-elle de détecter ?

Exercice 8. (**) Numéro ISBN

Norme ISO 2108 à partir du 1 janvier 2007

Identifie les livres par EAN-13 : les trois premiers chiffres valent 978, les 9 suivants sont les 9 premiers chiffres du code ISBN (ABC) et le dernier est le chiffre de contrôle EAN-13.

- Vérifier que le code 9782807335363 est valide.
- Compléter la séquence 210-059-911 en un numéro ISBN correct.
- Quel est son code-barres EAN-13 associé ?