EPFL - Information, Calcul et Communication - Partie théorique

Anciens examens

1 Algorithmique

Exercice 1 (automne 2020-2021).

a) Ecrire un algorithme **test**(L, n), *de complexité temporelle* $\Theta(n)$, qui prenne en entrée une liste L de nombres entiers ainsi que sa taille n, et dont la sortie soit oui si et seulement si la liste L est ordonnée dans l'ordre croissant *ou* dans l'ordre décroissant.

Exemples: Si $L = (2, 2, 2, 3, 8)$ ou $L = (10, 5, 4, 4, 2)$ (avec ici $n = 5$), alors la sortie doit être oui. Si par contre $L = (1, 4, 3, 7)$ (avec $n = 4$), alors la sortie doit être non.

b) On considère maintenant une liste L de taille n pour laquelle l'algorithme de la partie **a)** produit le résultat **test**$(L, n) =$ oui.

Ecrire un algorithme **tri**(L, n), *de complexité temporelle au plus* $\Theta(n)$, qui prenne en entrée la liste L ainsi que sa taille n, et dont la sortie soit une version de cette liste triée dans l'ordre croissant.

c) En général, il n'existe pas d'algorithme de complexité temporelle $\Theta(n)$ qui permette de trier une liste L quelconque de taille n. Ceci est dû au fait que, contrairement à la situation particulière de la partie **b)**, les éléments d'une liste L quelconque peuvent être ordonnés d'un grand nombre de façons. Par exemple, si L est une liste composée des 3 éléments 2, 4 et 7, alors ces éléments peuvent être ordonnés de 6 façons différentes:

$$L = (2, 4, 7) \quad L = (2, 7, 4) \quad L = (4, 2, 7) \quad L = (4, 7, 2) \quad L = (7, 2, 4) \quad \text{ou} \quad L = (7, 4, 2)$$

Si maintenant L est une liste composée de n éléments différents, de combien de façons différentes peut-on les ordonner?

d) Est-ce que $\Theta(n \cdot \log_2(n))$ bits suffisent pour représenter tous ces ordres différents? Justifier.

Exercice 2 (printemps 2020-2021).

On considère l'algorithme récursif suivant:

algo_rec

entrée : *nombre entier positif* $N \geq 1$
sortie : *??*

Si $N = 1$
  sortir : 1

Si N *est pair*
  sortir : **algo_rec**$(N/2)$

Sinon
  sortir : $1 +$ **algo_rec**$((N-1)/2)$

a) Que vaut la sortie de cet algorithme lorsque $N = 10$ en entrée ?

b) Pour une valeur quelconque de N en entrée, que représente la sortie de cet algorithme ?

c) Quelle est la complexité temporelle de cet algorithme? (utiliser la notation $\Theta(\cdot)$)

d) Proposer une version itérative de cet algorithme.

Exercice 3a (automne 2021-2022).

a) Soit L une liste de nombres entiers positifs tous différents les uns des autres et ordonnés dans l'ordre croissant; soit également n la taille de la liste L et x un nombre entier positif.

Ecrire un algorithme **algo1**(L, n, x), *de complexité temporelle* $\Theta(n)$, qui prenne en entrée la liste L, sa taille n ainsi que le nombre x, et dont la sortie soit le nombre de paires $\{i, j\} \subset \{1, \ldots, n\}$ avec $i < j$ telles que $L(i) + L(j) = x$.

Exemple: Si $L = (2, 4, 6, 8, 9, 10)$, $n = 6$ et $x = 14$, alors la sortie de **algo1**(L, n, x) doit être 2.

b) Supposons maintenant que les nombres de la liste L ne soient pas forcément tous différents les uns des autres, ni triés dans l'ordre croissant. Proposez un nouvel algorithme **algo2**(L, n, x) dont la sortie soit, dans ce cas plus général, la même que celle demandée au point a). Quelle est la complexité temporelle de votre algorithme? (utiliser la notation $\Theta(\cdot)$)

Exercice 3b (automne 2021-2022).

On considère l'algorithme suivant:

algo

entrée : *n nombre entier positif ou nul*
sortie : *???*

```
Si n = 0
    Sortir : 1
s ⟵ 0
Pour k allant de 0 à n − 1
    s ⟵ s + algo(k)
Sortir : s
```

a) Pour une valeur d'entrée $n \geq 1$, quelle est la sortie de cet algorithme?

b) Quelle est la complexité temporelle de cet algorithme? (utiliser la notation $\Theta(\cdot)$)

Exercice 4 (printemps 2021-2022).

a) Écrire un algorithme **algo1**(L, n) qui prenne en entrée une liste L de n nombres entiers relatifs (par exemple, $L = (+4, -12, +17, +3, -15, +34, -8, +16)$, pour $n = 8$) et dont la sortie soit la valeur maximale de la somme

$$\sum_{k=i}^{j} L(k)$$

parmi toutes les paires d'indices $i, j \in \{1, \ldots, n\}$ avec $i \leq j$ (dans l'exemple ci-dessus, la sortie doit donc être $+47$, valeur atteinte pour $i = 3$ et $j = 8$).

b) Quelle est la complexité temporelle de votre algorithme **algo1**(L, n) ?
(utiliser la notation $\Theta(\cdot)$)

On considère maintenant l'algorithme suivant:

algo2
entrée : *liste M de n nombres entiers relatifs* sortie : *???*
Pour *i allant de 1 à $n-1$* $\quad L(i) \longleftarrow M(i+1) - M(i)$ **Sortir : algo1**$(L, n-1)$

c) Que représente la sortie de l'algorithme **algo2**(M, n) pour une liste M donnée ?

d) Quelle est la complexité temporelle de l'algorithme **algo2**(M, n) ?
(utiliser la notation $\Theta(\cdot)$)

Exercice 5a (automne 2022-2023).

a) Soient L_1, L_2 deux listes de n nombres entiers positifs, chacune ordonnée dans l'ordre croissant. Ecrire un algorithme de complexité temporelle $\Theta(n)$ ou $\Theta(n \cdot \log_2(n))$ dont la sortie soit oui si et seulement s'il existe $i, j \in \{1, \ldots, n\}$ tels que $L_1(i) = L_2(j)$ [*Note:* on n'exclut pas ici le cas $i = j$].

b) Si maintenant la liste L_1 est de taille n et la liste L_2 est de taille 2^n (et chacune des deux listes est toujours ordonnée dans l'ordre croissant), existe-il un algorithme de complexité polynomiale en n dont la sortie soit oui si et seulement s'il existe $i \in \{1, \ldots, n\}$ et $j \in \{1, \ldots, 2^n\}$ tels que $L_1(i) = L_2(j)$? Si oui, écrivez un tel algorithme; si non, expliquez pourquoi ce n'est pas possible.

Exercice 5b (automne 2022-2023).

On considère l'algorithme suivant:

algorithme

entrée : *nombre entier strictement positif n*
sortie : *???*

Si $n = 1$
  Sortir : 1

Si *n est pair*
  Sortir : 1 + **algorithme**$(n/2)$

Sinon
  Sortir : algorithme$(n - 1)$

a) Quelle est la sortie de cet algorithme si $n = 32$ en entrée?

b) Quelle est la sortie de cet algorithme si $n = 31$ en entrée?

c) Pour une valeur quelconque de $n \geq 1$ en entrée, que représente la sortie de cet algorithme?

d) Quelle est la complexité temporelle de cet algorithme? (utiliser la notation $\Theta(\cdot)$)

Exercice 6 (printemps 2022-2023).

a) Ecrivez un algorithme qui prenne en entrée un nombre entier relatif x ainsi qu'une liste L de n nombres entiers relatifs et dont la sortie soit oui si et seulement s'il existe $i, j \in \{1, \ldots, n\}$ tels que $i < j$ et $L(i)+L(j) \geq x$. De plus, la complexité temporelle de votre algorithme doit être un $\Theta(n)$.

b) On considère maintenant le problème plus général suivant:

> "Etant donné un nombre entier relatif x et une liste L de n nombres entiers relatifs, existe-t-il un sous-ensemble $S \subset \{1, \ldots, n\}$ tel que $\sum_{j \in S} L(j) \geq x$?"

Ce problème fait-il partie de la classe NP ? Justifiez votre réponse.

c) Sait-on si le problème de la question b) fait également partie de la classe P ? A nouveau, justifiez votre réponse.

Exercice 7 (automne 2023-2024).

Considérons l'algorithme suivant:

mystère

entrée : *deux listes L_1, L_2 de nombres entiers positifs, toutes deux de taille $n = 2^k$ avec $k \geq 0$, et toutes deux ordonnées dans l'ordre croissant*
sortie : *nombre entier positif*

```
Si n = 1
    Sortir: (L1(1)+L2(1))/2
m ⟵ n/2
Si L1(m) > L2(m)
    Sortir: mystère(L1(1 : m), L2(m + 1 : n), m)
Sinon
    Sortir: mystère(L1(m + 1 : n), L2(1 : m), m)
```

a) Quelle est la sortie de l'algorithme **mystère**(L_1, L_2, n) si $L_1 = (1, 3, 6, 9)$, $L_2 = (2, 4, 7, 8)$ et $n = 4$ en entrée?

b) Est-ce que la sortie de l'algorithme est modifiée si on remplace en entrée la liste $L_1 = (1, 3, 6, 9)$ par $L_1 = (1, 3, 6, 355)$?

c) Quelle est la complexité temporelle de l'algorithme **mystère**(L_1, L_2, n)? (en notation $\Theta(\cdot)$)

d) Ecrire une version itérative (c'est-à-dire non-récursive) de l'algorithme **mystère**(L_1, L_2, n).

Exercice 8 (printemps 2023-2024).

Soient m, n deux nombres entiers positifs, A un tableau de $m \times n$ nombres entiers tels que

$$A(i,j) \leq A(i,\ell), \quad \text{pour tout} \quad 1 \leq i \leq m, \quad 1 \leq j < \ell \leq n$$

et

$$A(i,j) \geq A(k,j), \quad \text{pour tout} \quad 1 \leq i < k \leq m, \quad 1 \leq j \leq n$$

On considère l'algorithme suivant :

mystère

entrée : *A un tableau de $m \times n$ nombres entiers vérifiant les propriétés ci-dessus et x un nombre entier*
sortie : *valeur binaire (oui/non)*

```
i ⟵ 1
j ⟵ 1
Tant que i ≤ m et j ≤ n
    Si A(i,j) = x
        Sortir: oui
    Si A(i,j) < x
        j ⟵ j + 1
    Sinon
        i ⟵ i + 1
Sortir: non
```

a) Quelle est la sortie de l'algorithme **mystère** si les données d'entrée sont les suivantes ?

$$m = 3, \quad n = 4, \quad A = \begin{pmatrix} 13 & 32 & 40 & 100 \\ 12 & 17 & 39 & 65 \\ 5 & 14 & 31 & 38 \end{pmatrix} \quad \text{et} \quad x = 31$$

b) En général, dans quel cas la sortie de l'algorithme **mystère** est-elle un oui ?

c) Supposons que $m = n$ en entrée. Quelle est alors la complexité temporelle de l'algorithme **mystère** en fonction de n ? (utiliser la notation $\Theta(\cdot)$)

d) Supposons maintenant que $m = 2$ soit un nombre fixé. Quelle est alors la complexité temporelle de l'algorithme **mystère** en fonction de n ? (utiliser à nouveau la notation $\Theta(\cdot)$)

e) Dans ce dernier cas ($m = 2$), existe-t-il un algorithme dont la sortie soit identique à celle de l'algorithme **mystère**, mais dont la complexité temporelle soit *significativement* moindre (c'est-à-dire avec un ordre de grandeur inférieur en n) ? Si oui, écrire un tel algorithme ; si non, expliquer pourquoi un tel algorithme n'existe pas.

Exercice 9a (automne 2024-2025).

On considère l'algorithme suivant:

algo

entrée : *liste L de nombres entiers positifs, de taille n*
sortie : *???*

Si $n = 1$
  Sortir: $L(1)$

$m \longleftarrow \lfloor \frac{n}{2} \rfloor$
Sortir: $\frac{m}{n} \cdot$ **algo**$(L(1:m), m) + \frac{n-m}{n} \cdot$ **algo**$(L(m+1:n), n-m)$

Note: $L(a:b) = (L(a), L(a+1), \ldots, L(b-1), L(b))$ désigne la sous-liste de L composée des éléments compris entre l'indice a et l'indice b (compris).

a) Que vaut la sortie de l'algorithme **algo**(L, n) si $L = (8, 10, 5, 9)$ et $n = 4$ en entrée?

b) Que vaut la sortie de l'algorithme **algo**(L, n) pour une liste L de taille n en général?

c) Quelle est la complexité temporelle de **algo**(L, n)? (utiliser la notation $\Theta(\cdot)$ et justifier votre réponse; attention, car la réponse à la question n'est pas immédiate!)

Note: On considère ici que l'opération $\frac{m}{n} \cdot a + \frac{n-m}{n} \cdot b$ pour deux nombres a, b arbitraires compte comme une seule opération.

Supposons maintenant qu'on modifie l'algorithme précédent comme suit:

algobis

entrée : *liste L de nombres entiers positifs, de taille n*
sortie : *???*

Si $n = 1$
  Sortir: $L(1)$

$m \longleftarrow n - 1$
Sortir: $\frac{m}{n} \cdot$ **algobis**$(L(1:m), m) + \frac{n-m}{n} \cdot$ **algobis**$(L(m+1:n), n-m)$

d) Que vaut la sortie de ce nouvel algorithme **algobis**(L, n) en général? Justifier votre réponse.

e) Quelle est la complexité temporelle de ce nouvel algorithme **algobis**(L, n)? (utiliser à nouveau la notation $\Theta(\cdot)$ et justifier votre réponse)

Exercice 9b (automne 2024-2025).

a) Ecrire un algorithme qui prenne en entrée une liste L de nombres entiers relatifs, de taille n, dont la sortie soit oui si et seulement si

$$\sum_{i=1}^{k} L(i) \geq 0, \quad \forall 1 \leq k \leq n \tag{1}$$

et dont la complexité temporelle soit un $\Theta(n)$.

b) On considère maintenant l'algorithme suivant:

algo

entrée : *liste L de nombres entiers relatifs, de taille n*
sortie : *oui/non*

```
Pour i allant de 1 à n
    Si L(i) < 0
        Sortir : non
Sortir: oui
```

Ecrire un algorithme dont les entrées et sortie soient identiques à celui de la partie a), mais qui utilise explicitement l'algorithme **algo** ci-dessus comme sous-algorithme.

Note: Bien sûr, votre algorithme fera appel à **algo** avec une liste L différente de celle pour laquelle il cherche à savoir si la relation (1) de la page précédente est vérifiée ou non.

2 Représentation binaire

Exercice 10 (automne 2023-2024).

Considérons la représentation binaire des nombres entiers positifs sur 8 bits.

a) Dans cette représentation, que vaut le résultat de l'addition ci-dessous?

```
  0 1 0 1 0 1 0 1
  0 0 1 0 1 0 1 0
+ 0 0 0 1 0 1 0 1
```

b) Encore dans cette représentation, que vaut le résultat de la multiplication ci-dessous?

```
  0 1 0 1 0 1 0 1
x 0 0 0 0 0 0 1 1
```

c) Toujours dans cette représentation, soit x un nombre représenté par

$$x = b_7\, b_6\, b_5\, b_4\, b_3\, b_2\, b_1\, b_0$$

Le calcul de x^2 déclenche un overflow. Que peut-on en déduire sur la valeur des bits $b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0$?

Exercice 11 (automne 2024-2025).

a) Soit x le nombre entier positif représenté en binaire par 10010000. Quelle est la représentation binaire du nombre $x/2$?

b) Quelle est la représentation binaire du nombre $\sqrt{x}$?

c) En général, si x est un nombre entier positif pair représenté sur n bits, combien de bits seront-ils nécessaires pour représenter le nombre $x/2$?

d) En général, si x est un nombre entier positif représenté sur n bits, et $x = y^2$ avec y un nombre entier positif, combien de bits (approximativement) seront-ils nécessaires pour représenter le nombre y?

e) En considérant maintenant la représentation binaire des nombres entiers *relatifs* sur 8 bits, quel est le nombre représenté par 10010000?

f) Dans cette même représentation, est-ce que l'opération

```
  1 0 0 1 0 0 0 0
+ 1 0 0 1 0 0 0 0
```

mène à un overflow? Justifiez votre réponse.

g) Toujours dans cette même représentation, quel est le plus grand nombre entier positif qu'il est possible de représenter? (donner sa représentation décimale)

3 Circuits

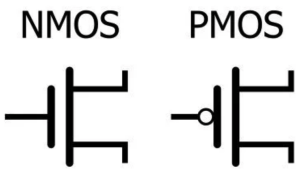
Exercice 12 (automne 2022-2023).

Construire un circuit logique avec des portes ET, OU et NON (et uniquement celles-ci), qui prend en entrée deux bits X et Y, et dont le bit de sortie S vaut 1 si et seulement si $X = Y = 1$ ou $X = Y = 0$.

Le maximum de 4 points est obtenu si votre circuit comporte au plus 4 portes (et que sa sortie est correcte). Chaque porte additionnelle entraîne une déduction de 1 point.

Exercice 13 (printemps 2023-2024).

En utilisant les transistors n-mos et p-mos vus au cours:



construire une porte NOR (=NON OU) dont la sortie s vaut 1 si et seulement si les deux entrées x et y valent chacune 0.

Rappel: Un transistor n-mos laisse passer le courant entre l'émetteur et le collecteur si et seulement si la tension d'entrée à la base est haute; il se passe exactement le contraire dans un transistor p-mos.

4 Signaux

Exercice 14 (automne 2020-2021).

On se donne un air de musique composé des notes suivantes, qu'on suppose être des sinusoïdes pures:

$$\text{SOL } (f_1 = 392 \text{ Hz}),\ \text{LA } (f_2 = 440 \text{ Hz}),\ \text{SI } (f_3 = 494 \text{ Hz}),\ \text{DO } (f_4 = 523.3 \text{ Hz}),\ \text{RÉ } (f_5 = 587.3 \text{ Hz})$$

L'air de musique est le suivant: (air connu)

$$\text{SOL SOL LA SOL DO SI} \quad \text{SOL SOL LA SOL RÉ DO}$$

La mélodie est donc composée de 12 notes en tout (l'espace n'est là que pour faire joli et donner une indication du rythme).

a) Si on passe cette mélodie à travers un filtre à moyenne mobile de période $T_c = 0,001$ seconde, va-t-on entendre des dissonances à la sortie du filtre? Justifiez votre réponse (sans équations).

b) Si on échantillonne maintenant cette mélodie à une fréquence $f_e = 1'000$ Hz, puis qu'on reconstruit le signal à l'aide de la formule d'interpolation, combien de notes seront reproduites correctement (parmi les 12 notes qui composent la mélodie)? Justifiez ce nombre.

c) En supposant qu'il n'est pas possible de changer la fréquence d'échantillonnage $f_e = 1'000$ Hz, proposez une méthode, en vous basant sur ce qu'on a vu en cours, pour éviter les dissonances lors de la reconstruction du signal. Décrivez en détail ce que vous faites.

d) On désire maintenant encoder cette séquence de notes sous la forme d'une séquence de bits. A priori, vu qu'il y a 5 notes différentes, on aurait besoin avec un encodage classique de 3 bits par note, donc de $12 \times 3 = 36$ bits en tout. Combien de bits *au plus* pouvez-vous économiser en utilisant une méthode optimale (qui respecte le principe d'encoder chaque note par une séquence de bits)? Décrivez en détail ce que vous faites.

e) Pour bien justifier l'utilisation de votre méthode, calculez l'entropie binaire de la séquence de notes en l'exprimant d'abord sous la forme $\frac{a}{b} + \frac{c}{d} \cdot \log_2(3)$, où a, b, c, d sont des nombres entiers, *puis* en calculant sa valeur approximative avec l'approximation $\log_2(3) \simeq 1.58$, et reliez ça avec ce que vous avez obtenu au point **d)**.

Exercice 15 (automne 2021-2022).

Un signal sonore $X = (X(t),\ t \in \mathbb{R})$ de bande passante $B = 6$ kHz est d'abord filtré par un filtre passe-bas idéal de fréquence de coupure f_c avant d'être échantillonné à une fréquence d'échantilonnage f_e, et chaque échantillon est ensuite encodé sur 32 bits.

a) Supposons que l'on souhaite enregistrer 50 minutes de ce signal sur un support mémoire d'une taille de 120 Mégaoctets (un octet représentant 8 bits). Quelles fréquences f_c et f_e choisir pour conserver au maximum les fréquences du signal X sans pour autant subir l'effet stroboscopique lors de la reconstruction du signal?

b) Si maintenant le signal X est donné par la formule explicite suivante:

$$X(t) = \sum_{n \geq 1} \frac{1}{n} \sin\left(\frac{2\pi f_0 t}{n}\right), \quad t \in \mathbb{R}$$

où $f_0 = 6$ kHz, quelle sera la formule pour le signal reconstruit Y dans le scénario précédent?

Exercice 16 (automne 2022-2023).

Rappel : Pour $a, b \in \mathbb{R}$, $\sin(a) - \sin(b) = 2 \cdot \sin\left(\frac{a-b}{2}\right) \cdot \cos\left(\frac{a+b}{2}\right)$.

On considère le signal $X(t) = \sin(2\pi f t)$, où f est une fréquence donnée (mesurée en Hz), ainsi que le signal $\widehat{X}(t) = X(t) - X(t - t_0)$, où $t_0 > 0$ est un intervalle de temps donné (mesuré en secondes).

a) Quelle est la bande passante du signal $\widehat{X}$?

b) Exprimer l'amplitude du signal $\widehat{X}$ en fonction de f et t_0.

c) Donner une valeur de t_0 (dépendant de f) pour laquelle l'amplitude du signal $\widehat{X}$ est maximale.

d) La transformation du signal X en un signal $\widehat{X}$ peut-elle être interprétée comme un filtre passe-bas? Justifier votre réponse.

Exercice 17 (printemps 2022-2023).

On considère un filtre un peu particulier dont l'effet est le suivant:

si le signal en entrée vaut $X(t) = a\ \sin(2\pi f t + \delta)$, alors le signal en sortie vaut $\widetilde{X}(t) = \frac{a}{f}\ \sin\left(2\pi f t + \frac{\delta}{f}\right)$.

A noter ici que f est la valeur de la fréquence mesurée en Hertz.

a) Argumentez pourquoi un tel filtre est un filtre passe-bas ou non.

b) Considérons maintenant le signal $X(t) = \cos(6\pi t) + \frac{1}{10}\sin(30\pi t)$. Que vaut le signal $\widetilde{X}(t)$?

Rappel: $\cos(x) = \sin(x + \frac{\pi}{2})$ pour tout $x \in \mathbb{R}$.

c) Pour rappel, l'amplitude d'une sinusoïde de fréquence f Hz passant à travers un filtre à moyenne mobile de période d'intégration T_c est diminuée d'un facteur

$$\left|\frac{\sin(\pi f T_c)}{\pi f T_c}\right|$$

Si on considère la composante $\frac{1}{10}\sin(30\pi t)$ du signal $X(t)$ ci-dessus comme du bruit, quel filtre permet de mieux atténuer cette composante: le filtre de la question a) (signal sortant $\widetilde{X}(t)$) ou un filtre à moyenne mobile de période d'intégration $T_c = 0.2$ sec (signal sortant $\widehat{X}(t)$ vu en cours) ? Justifiez votre réponse.

d) Si on échantillonne le signal $\widehat{X}(t)$ sortant du filtre à moyenne mobile à une fréquence $f_e = 20$ Hz, assistera-t-on à l'effet stroboscopique ? Justifiez votre réponse.

Exercice 18 (automne 2023-2024).

Rappel: Pour $a, b \in \mathbb{R}$, $\sin(a) \cdot \sin(b) = \frac{1}{2} \cdot (\cos(a-b) - \cos(a+b))$, $\sin(-a) = -\sin(a)$, $\cos(-a) = \cos(a)$.

a) Considérons les deux signaux X et Y suivants:

$$X(t) = \frac{1}{2} \cdot \cos(8\pi t) + \frac{1}{8} \cdot \sin(6\pi t) \quad \text{et} \quad Y(t) = \sin(5\pi t) \cdot \sin(3\pi t), \quad t \in \mathbb{R}$$

Chacun de ces signaux est filtré avec un filtre passe-bas idéal de fréquence de coupure $f_c = 3,9$ Hz.
Que valent les signaux filtrés $\widehat{X}$ et $\widehat{Y}$, et que valent leurs bandes passantes respectives $B_{\widehat{X}}$ et $B_{\widehat{Y}}$?

b) Considérons maintenant le signal $\widehat{Z}$ défini par

$$\widehat{Z}(t) = \widehat{X}(t) + \widehat{Y}(t), \quad t \in \mathbb{R}$$

Si ce signal est échantillonné avec une fréquence d'échantillonnage $f_e = 7$ Hz, assiste-t-on à l'effet stroboscopique lors de sa reconstruction ? Justifiez votre réponse.

c) Considérons finalement le signal Z défini par

$$Z(t) = X(t) + Y(t), \quad t \in \mathbb{R}$$

Si ce signal est échantillonné avec une fréquence d'échantillonnage $f_e = 7$ Hz, assiste-t-on à l'effet stroboscopique lors de sa reconstruction ? Justifiez votre réponse.

Exercice 19 (printemps 2023-2024).

Rappel: Si une sinusoïde pure $Y(t) = \sin(2\pi f t)$ passe à travers un filtre à moyenne mobile de période T_c, alors le signal sortant du filtre est donné par

$$\widehat{Y}(t) = \frac{\sin(\pi f T_c)}{\pi f T_c} \cdot \sin(2\pi f t - \pi f T_c)$$

a) On considère le signal $X(t) = \sin(6\pi t) + \sin(8\pi t)$. Quel est le signal $\widehat{X}(t)$ sortant après passage à travers un filtre à moyenne mobile de période $T_c = 0.25$ sec ?

b) Quelle est la bande passante du signal filtré $\widehat{X}$?

c) A quelle fréquence f_e faut-il échantillonner la signal $\widehat{X}$ si on désire éviter l'effet stroboscopique lors de sa reconstruction au moyen de la formule d'interpolation ?

d) Est-ce que la réponse à la question c) change si la période du filtre $T_c = 0.5$ sec ? Justifier.

Exercice 20 (automne 2024-2025).

Rappel: Pour $a, b \in \mathbb{R}$, $\sin(a) \cdot \sin(b) = \frac{1}{2} \cdot (\cos(a-b) - \cos(a+b))$, $\sin(-a) = -\sin(a)$, $\cos(-a) = \cos(a)$.

a) On considère le signal Z suivant:

$$Z(t) = \sin(4\pi t) \cdot \sin(3\pi t) - \sin(5\pi t) \cdot \sin(2\pi t), \quad t \in \mathbb{R}$$

Exprimez Z comme une somme de (co-)sinusoïdes et calculez sa bande passante.

b) Pour reconstruire un signal X à partir de sa version échantillonnée, on utilise la formule d'interpolation suivante:

$$X_I(t) = \sum_{m \in \mathcal{Z}} X\left(\frac{m}{2}\right) \cdot \mathrm{sinc}(2t - m), \quad t \in \mathbb{R}$$

où sinc est la fonction "sinus cardinal" vue au cours.

b1) Est-il toujours vrai que $X_I(n) = X(n)$ pour tout $n \in \mathbb{Z}$? Justifiez votre réponse.

b2) Quelle condition doit satisfaire la bande passante $f_{\max}$ du signal X pour garantir que $X_I(t) = X(t)$ pour tout $t \in \mathbb{R}$? A nouveau, justifiez votre réponse.

c) Le signal Z de la question a) ne satisfait malheureusement pas la condition trouvée à la question b2). Expliquez en détail ce qu'il faut faire avec le signal Z pour éviter l'effet stroboscopique lors de sa reconstruction, tout en conservant au mieux celui-ci. Que vaut alors le signal reconstruit?

5 Compression

Exercice 21 (printemps 2020-2021).

a) Ecrire un algorithme **algo** qui prenne en entrée deux listes de lettres L_1 et L_2, toutes deux de taille n, dont la sortie soit oui si et seulement s'il existe $i, j \in \{1, \ldots, n\}$ tels que $L_1(i) = L_2(j)$, et dont la complexité temporelle soit un $\Theta(n \cdot \log_2(n))$.

Note: Les algorithmes de recherche d'élément et de tri vus au cours peuvent s'utiliser avec des listes de lettres !

Dans la suite de cet exercice, $H(L)$ désigne l'entropie d'une liste de lettres L, et $L_1 \cup L_2$ désigne la liste de lettres qui est la concaténation des deux listes L_1 et L_2.

Exemple: Si $L_1 = (A, B, C)$ et $L_2 = (D, E, F)$, alors $L_1 \cup L_2 = (A, B, C, D, E, F)$.

b) Démontrer que si L_1 est une liste de taille n composée de lettres toutes différentes, L_2 est une autre liste de taille n, également composée de lettres toutes différentes, et si $\textbf{algo}(L_1, L_2, n) =$ non (comme dans l'exemple ci-dessus), alors

$$H(L_1 \cup L_2) = 1 + \frac{H(L_1) + H(L_2)}{2} \tag{2}$$

c) Comment la relation (2) change-t-elle si L_1, L_2 sont à nouveau deux listes de taille n composées chacune de lettres toutes différentes, mais si $\textbf{algo}(L_1, L_2, n) =$ oui ? (pas besoin ici de démonstration; vous pouvez vous aider d'un exemple pour répondre à la question)

Exercice 22 (automne 2021-2022).

Un texte est composé pour moitié de lettres tirées d'un alphabet de 16 lettres, et pour moitié d'espaces. On suppose de plus que les probabilités d'apparition de toutes les lettres sont égales, que les lettres et les espaces sont mélangés dans le texte, et que plusieurs espaces peuvent être consécutifs.

a) Proposez un système d'encodage de ce texte sous la forme d'une séquence de bits qui permette d'utiliser le plus petit nombre moyen de bits par caractère (lettre ou espace), tout en restant décodable de manière unique au moyen d'un dictionnaire; que vaut ce nombre?

b) Si maintenant le texte était composé pour deux tiers de lettres et pour un tiers d'espaces, l'encodage proposé en a) serait-il toujours optimal? Calculez également le nombre moyen de bits utilisés par caractère dans ce cas.

Exercice 23 (printemps 2021-2022).

Considérons la séquence de 20 lettres:

DABRACADABRACADABRAC

a) Calculer l'entropie de cette séquence, en l'exprimant tout d'abord sous la forme

$$H(X) = a + b\,\log_2(3) + c\,\log_2(5)$$

où a, b, c sont des fractions (positives ou négatives), *puis* en calculant sa valeur numérique approximative à l'aide des deux approximations $\log_2(3) \simeq 1.6$ et $\log_2(5) \simeq 2.3$.

b) Utiliser un algorithme de compression *optimal* vu au cours pour encoder cette séquence de lettres sous la forme d'une séquence de bits : dessiner l'arbre et noter le dictionnaire obtenu.

c) Combien de bits par lettre en moyenne votre algorithme utilise-t-il ? Votre résultat est-il cohérent avec la réponse obtenue à la question a) ? Justifier.

Pour transmettre le message ci-dessus et le protéger d'éventuelles erreurs qui pourraient survenir lors de la transmission, on vous propose maintenant trois méthodes différentes :

(I) utiliser le dictionnaire suivant: $A = 0000$, $B = 1110$, $C = 1101$, $D = 1011$ et $R = 0111$

(II) utiliser le dictionnaire suivant: $A = 000000$, $B = 111100$, $C = 001111$, $D = 101010$ et $R = 010101$

(III) prendre la séquence de bits produite à la question b), la découper en sous-séquences de 4 bits, et utiliser le codage de Hamming pour chacune de ces sous-séquences.

d) Quel est le nombre total de bits produits par chacune de ces trois méthodes lors de l'encodage de la séquence DABRACADABRACADABRAC (identique à celle du dessus)?

On suppose maintenant qu'au plus une erreur (de type: un 0 qui devient un 1, ou le contraire) se produit sur une séquence de 8 bits consécutifs.

e) Laquelle ou lesquelles des trois méthodes ci-dessus permettent-elles de *détecter* ces erreurs ? Justifier.

f) Laquelle ou lesquelles des trois méthodes ci-dessus permettent-elles de *corriger* ces erreurs ? Justifier.

Exercice 24 (automne 2022-2023).

On considère la séquence de 36 bits suivante :

000001010100 000010001100 000100010001

(les espaces ci-dessus sont juste ajoutés pour la lisibilité de la séquence, mais sont à ignorer)

Pour compresser cette séquence, on considère deux stratégies possibles :

1. Regrouper les bits par groupes de 4, en considérant chaque groupe de 4 bits comme un seul symbole, et appliquer l'encodage de Huffman à ces symboles.
2. Regrouper les bits par groupes de 3, en considérant chaque groupe de 3 bits comme un seul symbole, et appliquer l'encodage de Huffman à ces symboles.

Etablir le code de Huffman dans chacun des deux cas, et déterminer quel code mène à une meilleure compression de la séquence d'origine.

Exercice 25 (printemps 2022-2023).

Pour encoder sous forme binaire une séquence de 8 lettres, on utilise le dictionnaire suivant:

A	B	C	D
111	110	10	0

a) Sachant que le dictionnaire ci-dessus a été établi à l'aide de l'algorithme de Huffman, quelle peut être la séquence de 8 lettres ?

b) Combien de bits par lettre en moyenne sont-ils utilisés avec ce dictionnaire ?

Note: Vous pouvez laisser le résultat sous forme de fraction irréductible.

On désire maintenant modifier le dictionnaire ci-dessus pour encoder une séquence composée des mêmes 8 lettres, ainsi que de 8 fois la lettre E (donc la nouvelle séquence est composée de 16 lettres en tout).

c) Proposez un nouveau dictionnaire pour les lettres A, B, C, D et E, respectant les conditions suivantes:

1. Chaque nouveau mot de code doit contenir l'ancien mot de code (p.ex., un nouveau mot de code pour la lettre A pourrait être 0111, 1110, 1111 ou encore 01111 etc.).

2. Le dictionnaire doit être optimal pour l'encodage de la séquence de 16 lettres, et la séquence de bits ainsi produite doit être uniquement décodable.

d) Combien de bits par lettre en moyenne sont-ils utilisés avec ce nouveau dictionnaire ?

Note: A nouveau, vous pouvez laisser le résultat sous forme de fraction irréductible.

Exercice 26 (automne 2023-2024).

a) Dans cet exercice, il vous est demandé de proposer un dictionnaire pour encoder la séquence de 27 lettres

A A A B B B A A A B B B A A A B B B C C C D D D E F G

au moyen de symboles pris dans l'ensemble $\{0, 1, 2\}$ (qu'on appelle ci-dessous des "trits").

Votre dictionnaire doit respecter les trois conditions suivantes:

1. à chaque lettre de la séquence ci-dessus doit correspondre un unique mot de code;

2. la séquence de trits ainsi produite doit être uniquement décodable;

3. le nombre total de trits utilisés doit être minimal.

b) Combien de trits par lettre en moyenne votre code utilise-t-il ? (donner le résultat sous forme de fraction irréductible)

c) Pour cette même séquence de lettres, de combien de trits par lettre auriez-vous besoin si la troisième condition ci-dessus était remplacée par:

3'. tous les mots de code du dictionnaire doivent contenir le même nombre de trits ?

Justifiez votre réponse.

Exercice 27 (printemps 2023-2024).

Soit un message formé de 2^n lettres (avec $n \geq 3$), contenant 2^{n-2} fois la lettre A, 2^{n-2} fois la lettre B et 2^{n-3} fois la lettre C, le reste étant composé d'espaces (qu'on considère ici comme des lettres à part entière).

a) On encode ce message en une suite de 0 et de 1 en utilisant l'algorithme de Huffman. Ecrire ci-dessous le dictionnaire obtenu et dessiner l'arbre correspondant.

b) Calculer le nombre moyen de bits par lettre utilisé par cet encodage.

c) En utilisant l'approximation $\log_2(3) \simeq 1.6$, calculer (approximativement) l'entropie du message.

Exercice 28 (automne 2024-2025).

a) Calculez les trois entropies suivantes:

$$H(\text{HONO}) \qquad H(\text{LULU}) \qquad H(\text{HONOLULU})$$

b) Soient X et Y deux séquences composées chacune de n lettres, d'entropies respectives $H(X)$ et $H(Y)$ et telles qu'aucune lettre de la séquence X ne se retrouve dans la séquence Y (comme dans l'exemple ci-dessus avec $X = \text{HONO}$ et $Y = \text{LULU}$).

Soit aussi XY la séquence composée de la juxtaposition des deux séquences X et Y (HONOLULU dans l'exemple ci-dessus). Dans le cas général, exprimez l'entropie $H(XY)$ de la séquence XY en fonction des entropies $H(X)$ et $H(Y)$.

Note: Si vous êtes bloqués sur cette question, un conseil: passez à la suivante (question c) et revenez après.

c) Supposons maintenant que les séquences X et Y de la question b) soient encodées sous forme binaire, chacune avec un dictionnaire sans préfixe.

Proposez une méthode systématique pour obtenir un dictionnaire sans préfixe pour encoder la séquence XY. Illustrez votre méthode avec les séquences HONO, LULU et HONOLULU de la question a).

Note: Cette question peut aussi vous donner un indice sur la formule à obtenir pour la question b).

6 Correction d'erreurs

Exercice 29 (automne 2021-2022).

Afin d'envoyer un message sur internet, Alice décompose d'abord celui-ci en deux paquets de données X_1, X_2, composés chacun d'une séquence de 1024 bits. Alice envoie ensuite X_1 et X_2, ainsi que deux copies $X_3 = X_1$ et $X_4 = X_2$ de ces paquets, et également un cinquième paquet $X_5 = X_1 \oplus X_2$ (ce qui signifie que chaque bit de X_5 est l'addition modulo 2 des bits des paquets X_1 et X_2).

a) Bob est le destinataire de ces paquets, mais malheureusement, certains d'entre eux se perdent en cours de route... Combien de pertes de paquets Bob peut-il tolérer, dans le pire des cas, tout en restant sûr de pouvoir retrouver le message orignal X_1, X_2 envoyé par Alice? Expliquez votre raisonnement.

b) Supposons maintenant qu'Alice et Bob se mettent d'accord à l'avance sur une clé secrète K, dont les bits sont tirés uniformément au hasard. Expliquez comment Alice et Bob peuvent utiliser cette clé pour échanger les cinq paquets décrits ci-dessus, de sorte que si Eve intercepte même tous les paquets, elle ne puisse retrouver aucune information à propos de X_1 ou X_2. En particulier, quelle doit être la longueur minimum de la clé K (en nombre de bits)?

Exercice 30 (printemps 2022-2023).

Pour envoyer à Bob un message encodé initialement sur 3 bits x_1, x_2, x_3, Alice envoie le message suivant composé de 7 bits:

$$x_1, \quad x_2, \quad x_3, \quad x_1 \oplus x_2, \quad x_1 \oplus x_3, \quad x_2 \oplus x_3, \quad x_1 \oplus x_2 \oplus x_3$$

où $x \oplus y$ est l'opération XOR vue en cours.

a) Quelle est la distance minimale d de ce code correcteur d'erreurs ? Justifiez pleinement votre réponse !

b) En conséquence, combien d'effacements et d'erreurs un tel code permet-il de corriger ?

c) En particulier, si Bob reçoit le message 0001111, peut-il en déduire les valeurs des bits x_1, x_2, x_3 ? Justifiez votre réponse.

Exercice 31 (automne 2023-2024).

Pour envoyer à Bob un message x_1, x_2, x_3, x_4 encodé initialement sur 4 bits, Alice envoie le message suivant composé de 8 bits:

$$x_1, \quad x_2, \quad x_3, \quad x_4, \quad x_1 \oplus x_2 \oplus x_3, \quad x_1 \oplus x_2 \oplus x_4, \quad x_1 \oplus x_3 \oplus x_4, \quad x_2 \oplus x_3 \oplus x_4$$

où $x \oplus y$ est l'opération XOR vue en cours.

a) Quelle est la distance minimale d de ce code correcteur d'erreurs ? Justifiez votre réponse !

b) En conséquence, combien d'effacements et d'erreurs un tel code permet-il de corriger ? (donner des réponses *numériques*)

c) Quelle est la distance *maximale* entre deux mots de ce code binaire ? Donner deux mots de code à cette distance l'un de l'autre.

Exercice 32 (printemps 2023-2024).

On considère le code correcteur d'erreurs suivant : pour tout envoi de 4 bits x_1, x_2, x_3, x_4, ceux-ci sont protégés par les 3 bits de parité suivants :

$$x_5 = x_1 \oplus x_2, \quad x_6 = x_3 \oplus x_4, \quad x_7 = x_5 \oplus x_6$$

a) Quelle est la distance minimale de ce code correcteur d'erreurs ? Justifier.

b) Combien d'effacements et d'erreurs un tel code permet-il de corriger ? (donner des réponses *numériques*)

c) Si on sait qu'au plus une erreur est survenue lors de la transmission et qu'on reçoit la séquence 1111011, que sait-on alors sur le ou les mots de code envoyés à l'origine (donner toutes les possibilités) ?

Exercice 33 (automne 2024-2025).

Pour indiquer à Bob le temps qu'il fait chez elle, Alice utilise le code binaire suivant:

Il fait beau: 00000 Il pleut: 11111 Il neige: 10101 Il vente: 11100 Il fait froid: 00111

a) Calculez la distance minimale de ce code binaire, ainsi que le nombre d'effacements / d'erreurs qu'il peut corriger, dans le pire des cas.

b) Supposons qu'on sache qu'*au plus* une erreur (de type $0 \to 1$ ou $1 \to 0$) survienne lors d'une transmission. Est-il toujours possible pour Bob de *détecter* une telle erreur? Justifiez votre réponse.

c) Toujours sous la même hypothèse qu'à la question b), si Bob reçoit la séquence 00100, peut-il en déduire le temps qu'il fait chez Alice? Si oui, quel est-il? Si non, expliquez pourquoi.