Exemples de questions à choix multiples

Notez bien qu'*une seule* réponse est valable pour chaque question!

Question 1. Un algorithme a besoin d'environ n^2 opérations pour calculer une fonction $f(n)$ donnée. Sachant que l'algorithme calcule $f(20)$ en 2 heures, combien de temps environ l'algorithme prendra-t-il pour calculer $f(60)$?

(A) 6 heures (B) 9 heures (C) 18 heures (D) 400 heures

Question 2. Un algorithme avec paramètre d'entrée n effectue de l'ordre de 2^n opérations lors de son exécution. On observe qu'il s'exécute en une heure environ lorsque le paramètre d'entrée vaut $n = 10$. En combien d'heures l'algorithme s'exécutera-t-il (environ) si le paramètre d'entrée vaut $n = 40$?

(A) quatre heures (B) mille heures (C) un million d'heures (D) un milliard d'heures

Question 3. On considère l'algorithme suivant:

```
algorithme
entrée : liste L de nombre entiers, de taille n
sortie : deux nombres entiers m et M

m ⟵ min(L(1), L(2))
M ⟵ max(L(1), L(2))
Pour i allant de 3 à n
    Si L(i) < m
        M ⟵ m
        m ⟵ L(i)
    Sinon
        Si L(i) < M
            M ⟵ L(i)
Sortir : m, M
```

Que représentent les nombres m et M à la sortie de cet algorithme?

(A) les deux plus grands éléments de la liste L

(B) les deux plus petits éléments de la liste L

(C) le plus petit et le plus grand élément de la liste L

(D) le premier et le dernier élément de la liste L

Question 4. Trouver une paire avec la plus grande différence dans une liste de n nombres triés peut être fait au minimum en

(A) $\Theta(1)$ (B) $\Theta(\log_2(n))$ (C) $\Theta(n)$ (D) $\Theta(n \cdot \log_2(n))$

Question 5. Soit L une liste quelconque de n nombres entiers relatifs. Pour chacun des problèmes suivants, on suppose qu'on utilise l'algorithme le plus efficace possible pour le résoudre. Lequel nécessitera le *moins* d'opérations à effectuer lorsque n devient grand?

(A) Trier la liste L.

(B) Identifier s'il existe $j, k \in \{1, \ldots, n\}$ tels que $j \neq k$ et $L(j) + L(k) = 0$.

(C) Trouver la plus grande différence entre deux éléments de la liste.

(D) Identifier si au moins deux nombres de la liste sont identiques.

***

On considère l'algorithme avec les entrées et sorties suivantes:

algorithme
entrée : *liste L de nombres entiers, de taille n* sortie : *a et b, deux nombres entiers de la liste L tels que $\lvert b-a\rvert$ soit le plus petit possible*
...

Exemple: Si la liste est $L = \{2, -17, 8, 18, 15\}$ (avec $n = 5$), alors on veut que $a = 15$ et $b = 18$ (ou $a = 18$ et $b = 15$; l'ordre n'importe pas ici).

Question 6. Quelle série d'instructions remplaçant les ... ci-dessus produit la bonne sortie pour l'algorithme?

(A)
$a \longleftarrow 1$
$b \longleftarrow 2$
Pour i allant de 1 à $n-1$
  Pour j allant de $i+1$ à n
    Si $|L(j)-L(i)| < |L(b)-L(a)|$
      $a \longleftarrow i$
      $b \longleftarrow j$
Sortir : a, b

(B)
$a \longleftarrow L(1)$
$b \longleftarrow L(2)$
Pour i allant de 1 à $n-1$
  Pour j allant de $i+1$ à n
    Si $L(j) - L(i) < b - a$
      $a \longleftarrow L(i)$
      $b \longleftarrow L(j)$
Sortir : a, b

(C)
$L' \longleftarrow$ **tri par fusion**(L)
$a \longleftarrow L'(1)$
$b \longleftarrow L'(2)$
Pour i allant de 2 à $n-1$
  Si $L'(i+1) - L'(i) < b - a$
    $a \longleftarrow L'(i)$
    $b \longleftarrow L'(i+1)$
Sortir : a, b

(D)
$L' \longleftarrow$ **tri par fusion**(L)
$j \longleftarrow 1$
Pour i allant de 2 à $n-1$
  Si $L'(i+1) - L'(i) < L'(j+1) - L'(j)$
    $j \longleftarrow i$
Sortir : $L'(j), L'(j)+1$

[Note: On suppose que l'algorithme de **tri par fusion** ci-dessus trie la liste L dans l'ordre croissant.]

Question subsidiaire. Quelle est la complexité temporelle de cet algorithme?

Question 7. On considère l'algorithme suivant:

algo2
entrée : *nombres entiers positifs* m, n sortie : *oui ou non*

$m \longleftarrow m(m+1)/2$
$n \longleftarrow n(n+1)/2$
Si $n \geq 2m$
    Sortir : *oui*
Sinon
    Sortir : *non*

Laquelle des affirmations ci-dessous est *fausse*?
(on rappelle ici que m et n sont les données en *entrée* de l'algorithme)

(A) Si $n = 6$ et $m = 4$, alors la sortie de cet algorithme est non.

(B) Si $n = 5$ et $m = 3$, alors la sortie de cet algorithme est oui.

(C) Si $n = 4$ et $m = 3$, alors la sortie de cet algorithme est non.

(D) Si $n = 4$ et $m = 2$, alors la sortie de cet algorithme est oui.

Question 8. On considère l'agorithme suivant:

algo3
entrée : *nombre entier positif* n sortie : *nombre entier positif* s

$i \longleftarrow 1$
$s \longleftarrow 0$
Tant que $i < 2^n$
    $s \longleftarrow s + i$
    $i \longleftarrow 2i$
Sortir : s

Quelle est la sortie de l'algorithme ci-dessus si l'entrée $n = 3$?

(A) $s = 15$ (B) $s = 7$ (C) $s = 6$ (D) $s = 3$

Question 9. Quel est la complexité temporelle de l'algorithme ci-dessus?

(A) $\Theta(2^n)$ (B) $\Theta(n^2)$ (C) $\Theta(n)$ (D) $\Theta(\log_2(n))$

On considère l'algorithme suivant:

algo
entrée : *entier naturel* n sortie : *??*

Si $n = 0$
    sortir : 0
Si $n = 1$
    sortir : 1
sortir : **algo**$(n-1)-$ **algo**$(n-2)$

Question 10. Parmi les affirmations suivantes, laquelle est vraie?

(A) Si n est pair, alors la sortie de l'algorithme est égale à $+1$.

(B) Si n est un multiple de 3, alors la sortie de l'algorithme est égale à 0.

(C) Si n est un multiple de 5, alors la sortie de l'algorithme est égale à -1.

(D) La sortie de l'algorithme peut être différente de -1, 0 ou $+1$.

Question 11. Quel est la complexité temporelle de l'algorithme ci-dessus?

(A) $\Theta(2^{n/2})$ ou plus (B) $\Theta(n^2)$ (C) $\Theta(n)$ (D) $\Theta(\log_2 n)$

***

Question 12. On considère l'algorithme suivant:

algo_rec1
entrée : *nombres entiers* $m \geq 1$ *et* $n \geq 0$ sortie : *???*

Si $n \geq m$
    Sortir : algo_rec1$(m, n-m)$
Sinon
    Sortir : n

Laquelle des affirmations ci-dessous est vraie?

(A) La sortie de cet algorithme est le reste de la division de n par m.

(B) La sortie de cet algorithme est le pgdc (plus grand diviseur commun) de m et n.

(C) La sortie de cet algorithme vaut n si $n = m$.

(D) Cet algorithme récursif ne s'arrête jamais, car il n'a pas de condition de terminaison.

Question 13. On considère l'algorithme suivant:

algo_rec2
entrée : *liste non-vide L de nombres entiers*
sortie : *???*

$n \longleftarrow taille(L)$
Si $n = 1$
  Sortir : $L(1)$
Si $L(n) \geq L(1)$
  Sortir : algo_rec2$(L(2:n))$
Sinon
  Sortir : algo_rec2$(L(1:n-1))$

Laquelle des affirmations ci-dessous est vraie?

(A) Si $L = \{6, 5, 4, 3, 2, 1\}$, alors la sortie de cet algorithme vaut 1.

(B) Si $L = \{1, 2, 3, 4, 5, 6\}$, alors la sortie de cet algorithme vaut 1.

(C) Si $L = \{6, 6, 5, 5, 4, 4\}$, alors la sortie de cet algorithme vaut 4.

(D) Si $L = \{1, 1, 2, 2, 3, 3\}$, alors la sortie de cet algorithme vaut 3.

Question 14. *Rappel:* On dit qu'un graphe composé de sommets et d'arêtes est *coloriable avec x couleurs* s'il est possible de colorier chaque sommet du graphe avec une de ces x couleurs de telle façon que toute paire de sommets reliés par une arête commune aient une couleur différente.

Lequel des problèmes suivants est le plus difficile?

(A) Construire un graphe avec n sommets et au moins n arêtes qui soit coloriable avec 2 couleurs.

(B) Construire un graphe avec n sommets et au moins n arêtes qui soit coloriable avec 3 couleurs.

(C) Etant donné un graphe avec n sommets et au moins n arêtes, décider si celui-ci est coloriable avec 2 couleurs.

(D) Etant donné un graphe avec n sommets et au moins n arêtes, décider si celui-ci est coloriable avec 3 couleurs.

Question 15. On considère le problème suivant:

Etant donné un graphe simple avec n sommets, on désire colorier les *arêtes* du graphe avec 5 couleurs différentes, de telle façon à ce que deux arêtes partageant un sommet commun soient toujours coloriées avec des couleurs différentes.

[Note: Un graphe *simple* est un graphe dont deux sommets sont reliés par *au plus* une arête.]

Laquelle des affirmations ci-dessous est-elle correcte?

(A) Vu que le nombre d'arêtes du graphe est potentiellement beaucoup plus grand que n, ce problème n'est pas dans NP.

(B) Vu que le nombre de couleurs est plus grand ou égal à 3, on sait que ce problème n'est pas dans P.

(C) Ce problème est dans NP.

(D) Ce problème est dans P, et n'est donc pas dans NP.

Question 16. On considère le problème suivant:

“Etant donnés C et n deux nombres entiers positifs et L une liste quelconque de n nombres entiers positifs, existe-t-il une séquence de n bits $(x(1), \ldots, x(n))$ telle que

$$\sum_{i=1}^{n} x(i) \cdot L(i) = C \text{ ? ”}$$

Laquelle des affirmations suivantes est vraie ?

(A) On sait que ce problème est dans NP.
(B) On ne sait pas si ce problème est dans NP.
(C) On sait que ce problème est dans P.
(D) On sait que ce problème n'est pas dans P.

Question 17. Quel est le principal *défaut* de l'algorithme ci-dessous ?

algo
entrée : *nombre entier positif* n sortie : *nombre entier positif* m

$m \longleftarrow 1$
Si $n = 1$
    Sortir : m
Pour i *allant de* 1 *à* $\mathbf{n-1}$
    $m \longleftarrow m + \mathbf{algo}(i)$
Sortir : m

(A) Sa sortie est exponentielle en n.
(B) Il recalcule plusieurs fois la même chose, ce qui implique ici que sa complexité est exponentielle en n.
(C) Il ne s'arrête jamais.
(D) Il est récursif.

Question subsidiaire. Et si on remplace le $\mathbf{n-1}$ en gras dans la boucle de l'algorithme par $\mathbf{n}$, quelle est la bonne réponse à la question ci-dessus ?

Question 18. Si $x = 00000100$ dans la représentation binaire des entiers positifs sur 8 bits, alors que vaut $2x - 1$ dans cette même représentation ?

(A) 00000111 (B) 00001001 (C) 00001111 (D) 00001011

Question 19. Toujours avec la même représentation, et avec la même valeur de x, l'opération $x + y$ déclenche un overflow. Que peut-on en conclure ?

(A) y est plus petit que x (B) $01000000 \leq y \leq 01111111$

(C) $y = 11111011$ (D) $y \geq 11111100$

Question 20. Sur la planète Czar, où vivent les plus et les minus, le jeu de diffball est très populaire. Le score final d'un match de diffball est simplement le nombre de buts marqués par les plus, moins le nombre de buts marqués par les minus. Sachant qu'une équipe ne marque jamais plus de 100 buts durant un match, de combien de bits a-t-on besoin *au minimum* pour enregistrer le score d'un match ?

(A) 7 (B) 8 (C) 14 (D) 200

Question 21. On considère 2 signaux $(X(t),\ t \in \mathbb{R})$ et $(Y(t),\ t \in \mathbb{R})$ tels que la bande passante B_X du signal X est strictement plus grande que la bande passante B_Y du signal Y. Laquelle des affirmations ci-dessous est vraie?

(A) La bande passante du signal $X + Y$ vaut B_X.

(B) La bande passante du signal $X + Y$ vaut B_Y.

(C) La bande passante du signal Z défini par $Z(t) = \max(X(t), Y(t))$ pour $t \in \mathbb{R}$ vaut B_X.

(D) La bande passante du signal $X \cdot Y$ peut être plus grande que $B_X + B_Y$.

Question 22. On considère le signal suivant:

$$X(t) = \sin(2\pi f_0 t) \cdot \cos(4\pi f_0 t), \quad t \in \mathbb{R}$$

qu'on échantillonne à une fréquence f_e. Laquelle des conditions ci-dessous faut-il respecter pour garantir une reconstruction parfaite du signal à partir de sa version échantillonnée, tout en permettant d'utiliser la plus petite fréquence d'échantillonnage possible?

(A) $f_e > 2f_0$ (B) $f_e > 4f_0$

(C) $f_e > 6f_0$ (D) $f_e > 8f_0$

Question 23. Un signal audio $(X(t),\ t \in \mathbb{R})$ a une bande passante de 44 kHz. Lequel des traitements ci-dessous évite l'effet stroboscopique lors de la reconstruction du signal?

(A) On échantillonne le signal X à une fréquence de 44.1 kHz, puis on le filtre avec un filtre passe-bas idéal de fréquence de coupure 22 kHz.

(B) Comme on n'entend de toute façon pas les fréquences au-delà de 22 kHz, on échantillonne le signal X à une fréquence de 44.1 kHz sans le filtrer.

(C) On filtre le signal X avec un filtre passe-bas idéal de fréquence de coupure 20 kHz, puis on l'échantillonne à une fréquence de 22.05 kHz.

(D) On filtre le signal X avec un filtre passe-bas idéal de fréquence de coupure 8'800 Hz, puis on l'échantillonne à une fréquence de 44.1 kHz.

***

Question 24. Quelle est l'entropie de la séquence de lettres suivante?

PASTA ALL'ARRABBIATA (18 lettres, sans compter l'espace ni l'apostrophe)

(A) $-\frac{3}{9} + 2\log_2(3) - \frac{7}{18}\log_2(7)$ (B) $\frac{5}{9} + 2\log_2(3) - \frac{7}{18}\log_2(7)$

(C) $\frac{14}{9} + \log_2(3) - \frac{7}{18}\log_2(7)$ (D) $\frac{10}{9} + \log_2(3) - \frac{7}{18}\log_2(7)$

Question 25. On encode la phrase de la question précédente avec le code de Huffman. Combien de bits sont utilisés en tout?

(A) 47 (B) 49 (C) 51 (D) 53

Question 26. Soit $\mathcal{C}$ un code binaire obtenu à partir de l'algorithme de Huffman. Laquelle des affirmations ci-dessous ne peut *jamais* être vraie?

(A) Il existe un nombre entier $k \geq 1$ tel que $\mathcal{C}$ contient des mots de code de longueur k, d'autres de longueur $k+2$, mais pas de mots de longueur $k+1$.

(B) Il existe un nombre entier $k > 1$ et une suite de k bits $b_1, b_2, \ldots, b_{k-1}, b_k$ tel que $\mathcal{C}$ contient le mot de code $\mathbf{c}$ donné par la suite de bits $b_1 b_2 \ldots b_{k-1} b_k$, ainsi que le mot de code $\mathbf{c}'$ donné par la suite de bits $b_k b_{k-1} \ldots b_2 b_1$.

(C) Il existe un nombre entier $k > 1$ et une suite de k bits $b_1, b_2, \ldots, b_{k-1}, b_k$ tel que $\mathcal{C}$ contient le mot de code $\mathbf{c}$ donné par la suite de bits $b_1 b_2 \ldots b_{k-1} b_k$, ainsi que le mot de code $\mathbf{c}'$ donné par la suite de bits $b_2 b_3 \ldots b_{k-1} b_k$.

(D) Il existe un nombre entier $k > 1$ tel quel le mot de code le plus long appartenant à $\mathcal{C}$ a une longueur de k bits et le nombre de mots de code dans $\mathcal{C}$ est strictement plus grand que 2^k.

Question 27. Pour envoyer un message parmi les 8 messages suivants: N, NE, E, SE, S, SW, W et NW, on utilise un code correcteur d'erreur $\mathcal{C}$ dont les 8 mots de code sont tous distincts et ont une longueur de 4 bits chacun. Laquelle des affirmations ci-dessous est *toujours* vraie?

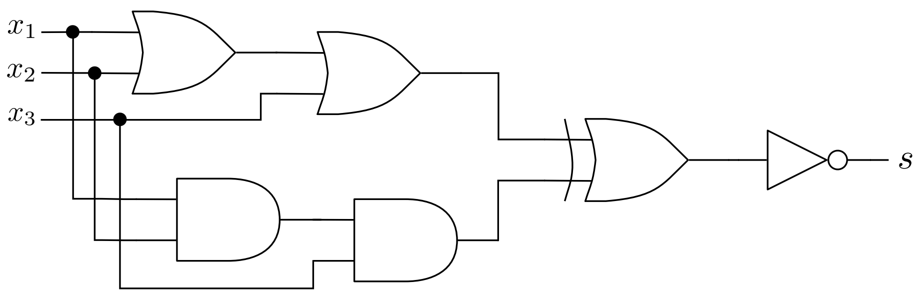
(A) Le code $\mathcal{C}$ permet de corriger une erreur.

(B) La distance minimale du code $\mathcal{C}$ est forcément plus petite ou égale à 2.

(C) La distance minimale du code $\mathcal{C}$ est forcément plus grande ou égale à 2.

(D) Le code $\mathcal{C}$ permet de corriger un effacement.

Question 28. On considère le circuit suivant:



Laquelle des affirmations ci-dessous est vraie?

(A) La sortie $s = 1$ si et seulement si $x_1 \oplus x_2 \oplus x_3 = 1$.

(B) La sortie $s = 1$ si et seulement si
$$(x_1 = 1 \text{ ET } x_2 = x_3 = 0) \text{ OU } (x_2 = 1 \text{ ET } x_1 = x_3 = 0) \text{ OU } (x_3 = 1 \text{ ET } x_1 = x_2 = 0)$$

(C) La sortie $s = 1$ si et seulement si x_1, x_2 et x_3 prennent tous des valeurs différentes.

(D) La sortie $s = 1$ si et seulement si $x_1 = x_2 = x_3$.

Question 29. La proposition logique $s = (x_1 \text{ ET } (\text{NON } x_2)) \oplus (x_1 \text{ OU } (\text{NON } x_2))$ peut s'écrire de manière simplifiée comme:

(A) $s = x_1$ (B) $s = \text{NON } (x_1 \oplus x_2)$

(C) $s = x_2$ (D) $s = x_1 \oplus x_2$

Réponses: 1.C, 2.D, 3.B, 4.A, 5.C, 6.C, 6sub.$\Theta(n \cdot \log_2(n))$, 7.A, 8.B, 9.C, 10.B, 11.A, 12.A, 13.D, 14.D, 15.C
16.A, 17.B, 17sub.C, 18.A, 19.D, 20.B, 21.A, 22.C, 23.D, 24.B, 25.B, 26.D, 27.B, 28.D, 29.B