Place : 0

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
EIDGENÖSSISCHE TECHNISCHE HOCHSCHULE – LAUSANNE
POLITECNICO FEDERALE – LOSANNA
SWISS FEDERAL INSTITUTE OF TECHNOLOGY – LAUSANNE

Faculté Informatique et Communications
Cours ICC aux sections MA et PH
Chappelier J.-C.

INFORMATIQUE, CALCUL & COMMUNICATIONS

Sections MA & PH

Examen intermédiaire I

25 octobre 2019

SUJET 1

Instructions :

- Vous disposez d'une heure quinze minutes pour faire cet examen (15h15 - 16h30).
- L'examen est composé de 2 parties : un questionnaire à choix multiples, à 12 points, prévu sur 40–45 minutes, et une partie à questions ouvertes, à 12 points, prévue sur 30–35 minutes. Mais vous êtes libres de gérer votre temps comme bon vous semble.
- **AUCUN DOCUMENT N'EST AUTORISÉ, NI AUCUN MATÉRIEL ÉLECTRONIQUE**.
- Vous devez **écrire à l'encre noire ou bleu foncée**, pas au crayon, ni en une autre couleur.
- Pour la première partie (questions à choix multiples), chaque question n'a qu'une seule réponse correcte parmi les quatre propositions. Il n'y a pas de point négatif.
  Indiquez vos réponses en bas de **cette** page en écrivant *clairement* pour chaque question une lettre majuscule parmi A, B, C et D. (Vous êtes autorisés à dégrafer cette page)
  Aucune autre réponse ne sera considérée, et en cas de rature, ou de toute ambiguïté de réponse, nous compterons la réponse comme fausse.
- Pour la seconde partie, répondez directement sur la donnée, à la place libre prévue à cet effet. Aucune feuille supplémentaire ne sera considérée.
- Toutes les questions comptent pour la note finale.

Notations : dans cet examen, le premier élément d'une liste L est noté $L(1)$.

Réponses aux quiz :

Reportez ici *en majuscule* la lettre de la réponse choisie pour chaque question, sans aucune rature.

1	2	3	4	5	6	7	8	9	10	11	12

PARTIE QUIZ

Question 1) Soit L une liste de nombres entiers relatifs. Lequel des problèmes suivants est le plus simple algorithmiquement (nécessite le moins d'opérations pour être résolu) ?

A] Trier L.

B] Déterminer s'il existe deux éléments de L dont la somme est égale à 0.

C] Trouver la plus petite différence entre deux éléments de L.

D] Déterminer s'il existe un sous-ensemble d'éléments de L dont la somme est égale à 0.

Question 2) On considère la machine de Turing dont la table de transition est :

	0	1	ε
1	(2, ε, +)	(4, ε, +)	(7, 1, −)
2	(2, 0, +)	(2, 1, +)	(3, ε, −)
3	(6, ε, −)	(7, 0, −)	(7, 1, −)
4	(4, 0, +)	(4, 1, +)	(5, ε, −)
5	(7, 0, −)	(6, ε, −)	(7, 1, −)
6	(6, 0, −)	(6, 1, −)	(1, ε, +)

Quel est l'état de la bande lorsque la machine s'arrête, si elle a démarré dans l'état 1 avec sa tête de lecture positionnée comme suit :

$$\cdots\varepsilon \quad \underset{\uparrow}{1} \quad 0 \quad 0 \quad 0 \quad 1 \quad \varepsilon\cdots$$

A] $\cdots\varepsilon\ 1\ 0\ 0\ 0\ \varepsilon\cdots$

B] $\cdots\varepsilon\ 1\ \varepsilon\cdots$

C] $\cdots\varepsilon\ 0\ 0\ 0\ \varepsilon\cdots$

D] $\cdots\varepsilon\ 0\ \varepsilon\cdots$

Question 3) Laquelle des représentations binaires suivantes sur 6 bits du nombre $e = 2.718...$ a la plus petite erreur ?

A] La représentation en virgule flottante avec 2 bits pour l'exposant et 4 bits pour la mantisse :

$$\widehat{x} = 1, m_1m_2m_3m_4 \times 2^{e_1e_2}$$

B] La représentation en virgule flottante avec 3 bits pour l'exposant et 3 bits pour la mantisse :

$$\widehat{x} = 1, m_1m_2m_3 \times 2^{e_1e_2e_3}$$

C] La représentation en virgule fixe avec 4 bits pour la partie fractionnaire et 2 bits pour la partie entière :

$$\widehat{x} = e_1e_2, f_1f_2f_3f_4$$

D] La représentation en virgule fixe avec 3 bits pour la partie fractionnaire et 3 bits pour la partie entière :

$$\widehat{x} = e_1e_2e_3, f_1f_2f_3$$

Question 4) On considère ici *uniquement* des schémas binaires sur 8 bits représentant des nombres entiers positifs. Avec

$$a = \texttt{00001011} \qquad \text{et} \qquad b = \texttt{00010101}\ ,$$

laquelle des opérations suivantes donne un résultat correct ?

A] $b \times b$ **B]** 2^a **C]** $a - b$ **D]** $a \times b$

suite au dos ☞

Question 5) Vous organisez un tournoi d'échecs en 8 tours (chaque joueur jouera 8 parties). Vous souhaitez stocker dans un programme, pour chaque joueur, la liste de chacun de ses opposants pendant le tournoi, ainsi que savoir s'il a joué avec les blancs ou les noirs. Sachant que vous avez 100 participants en tout (et que leur liste est déjà stockée par ailleurs), combien de bits vous faut-il prévoir *au minimum par joueur* pour stocker ces informations ?

A] 808 **B]** 16 **C]** 57 **D]** 64

Question 6) Considérer l'algorithme suivant, où $L(i:j)$ représente la sous-liste $(L(i), ..., L(j))$:

algo6
entrée : *L liste non vide de nombres réels*
sortie : *? ? ?*

$t \longleftarrow$**taille**(L)
Si $t = 1$
  Sortir : $(L(1), L(1))$
$L' \longleftarrow$**algo6**$(L(2:t))$
$L'' \longleftarrow$**algo6**$(L(1:t-1))$
Si $L(t) < L(1)$
  Sortir : $(L'(1), L''(2))$
Sinon
  Sortir : $(L''(1), L'(2))$

Quelle est sa sortie pour l'entrée $L = (12, 24, 33, 55, -3, 43, 8, 0)$?

A] $(12, 0)$ **B]** $(0, 12)$ **C]** $(33, 33)$ **D]** $(-3, 55)$

Question 7) Si n est la taille de la liste L en entrée, comment se situe la complexité de l'algorithme précédent ?

A] en $\mathcal{O}(n)$, mais pas en $\mathcal{O}(\log(n))$.
B] en $\mathcal{O}(\log(n))$.
C] en $\mathcal{O}(n^2)$, mais pas en $\mathcal{O}(n)$.
D] en $\mathcal{O}(2^n)$, mais pas en $\mathcal{O}(n^2)$.

Question 8) Considérer l'algorithme suivant, où $L(i:j)$ représente la sous-liste $(L(i), ..., L(j))$ si $i \leq j$, et la liste vide sinon :

algo8
entrée : *liste L*
sortie : *? ? ?*

$t \longleftarrow$ **taille**(L)
Si $t < 2$
  Sortir : 1
Si $L(1) = L(t)$
  Sortir : algo8$(L(2:t-1))$
Sinon
  Sortir : 0

algo8(7, 5, 12, 4, 8, 12, 5, 7) :

A] ne se termine pas. **B]** sort 0. **C]** ne sort rien du tout. **D]** sort 1.

Question 9) Un algorithme a besoin de l'ordre de 2^n instructions élémentaires pour calculer une fonction $f(n)$ donnée. Sachant qu'un programme implémentant (uniquement) cet algorithme sur une machine donnée calcule $f(50)$ en 10 minutes, combien de minutes environ faudrait-il à ce même programme pour calculer $f(200)$ (en repartant du début) ?

Note : on pourra si nécessaire faire l'approximation $2^{10} \simeq 10^3$.

A] 10^{60} **B]** 10^4 **C]** 10^{46} **D]** 40

Question 10) Quelle est la sortie de l'algorithme suivant :

algo10
entrée : *deux entiers a et b, strictement positifs*
sortie : *???*

```
n ⟵ 1
m ⟵ 1
Tant que n < b
    n ⟵ n + 1
    m ⟵ m × a
Sortir : n × m
```

A] $b \times a^{b-1}$ **B]** $(b-1) \times a^{b-1}$ **C]** $(b-1) \times a^b$ **D]** $b \times a^b$

Question 11) Avec $a = b$ en entrée, comment se situe la complexité de l'algorithme précédent ?

A] en $\mathcal{O}(1)$.

B] en $\mathcal{O}(b^2)$, mais pas en $\mathcal{O}(\log(b))$.

C] en $\mathcal{O}(2^b)$, mais pas en $\mathcal{O}(b^2)$.

D] en $\mathcal{O}(\log(b))$, mais pas en $\mathcal{O}(1)$.

Question 12) ⚠ Pour l'algorithme suivant :

algo12
entrée : *deux entiers positifs a et b*
sortie : *0 ou 1*

```
b ⟵ a × (a − 1)/2
a ⟵ b × (b + 1)/2
Si a ≤ 2b
    Sortir : 1
Sinon
    Sortir : 0
```

laquelle des propositions suivantes est vraie ?

A] Pour l'entrée $a = 2$, $b = 2$, il sort 0.

B] Pour l'entrée $a = 1$, $b = 3$, il sort 1.

C] Pour l'entrée $a = 5$, $b = 3$, il sort 1.

D] Aucune des autres propositions.

---

suite au dos ☞

PARTIE EXERCICES

1 – Ecriture d'algorithmes [12 points]

On s'intéresse ici à quelques opérations sur les représentations binaires de nombres entiers positifs ou nuls. Pour cela, on représentera les nombres en question sous forme de listes de valeurs 0 ou 1 telles que $L(1)$ soit le bit de poids le plus faible du nombre représenté et $L(n)$, où n est la taille de la liste, soit toujours à 1, sauf s'il s'agit de la liste vide, laquelle représente le nombre 0. Par exemple, la liste représentant le nombre 22 (dont l'écriture binaire usuelle est `10110`) sera $(0, 1, 1, 0, 1)$.

Question 13) **[1.5 points]** Ecrivez un algorithme qui, prenant en entrée une telle liste, sort la valeur décimale correspondante. Par exemple, pour la liste $(0, 1, 1, 0, 1)$ en entrée, cet algorithme devra sortir 22. Pour la liste vide, il sortira 0.

Question 14) **[1 point]** Déterminez la complexité de l'algorithme que vous avez écrit à la question précédente (question 13). Justifiez votre réponse.

Question 15) **[3 points]** Ecrivez un algorithme *récursif, sans aucune boucle ni itération*, qui, prenant en entrée deux listes binaires, sort le nombre de bits en commun, c.-à-d. le nombre de bits ayant la même valeur à la même position dans les deux listes.

Par exemple, pour les listes $(0, 1, 1, 0, 1)$ et $(0, 0, 1, 1)$ en entrée, cet algorithme devra sortir 2 car les premier et troisième bits de ces deux listes sont égaux.

A noter que les deux listes en entrée n'ont pas nécessairement la même taille.

Question 16) **[1.5 point]** Déterminez la complexité de l'algorithme que vous avez écrit à la question précédente (question 15) appliqué à des listes de même taille (n). Justifiez votre réponse.

Question 17) **[5 points]** Ecrivez un algorithme qui, prenant en entrée deux listes binaires, sort la liste correspondant à l'addition des deux nombres représentés.

Par exemple, pour les listes $(0, 1, 1, 0, 1)$ (représentant 22) et $(0, 0, 1, 1)$ (représentant 12) en entrée, cet algorithme devra sortir la liste $(0, 1, 0, 0, 0, 1)$ (représentant 34).

A noter que les deux listes en entrée n'ont pas nécessairement la même taille et que la liste en sortie peut être plus grande que la plus grande des deux en entrée.

Réponses :

(Utilisez cette page pour continuer de répondre à l'exercice ci-contre, mais veuillez s.v.p. préciser le numéro de la question traitée.)

(Utilisez cette page pour, si nécessaire, continuer de répondre à l'exercice, mais veuillez s.v.p. préciser le numéro de la question traitée.)