

INFORMATIQUE, CALCUL & COMMUNICATIONS

Sections MA & PH

Correction Examen intermédiaire III

19 décembre 2014

SUJET 1

Instructions :

- VOUS DEVEZ BIEN SÛR FAIRE CES CORRECTIONS DANS LA PLUS STRICTE CONFIDENTIALITÉ.

Veuillez en particulier ne *pas* diffuser le corrigé. Merci.

- Veuillez à garder l'ordre alphabétique de classement des copies.
- Re-vérifiez bien deux fois vos corrections. Nous faisons tous des erreurs...
- Une fois la correction terminée, remplissez le fichier de notes envoyé par email.
Mettre explicitement un 0 à tous ceux qui ont rendu une feuille blanche (par opposition à « vide » qui signifiera que vous n'avez pas vu de copie à ce nom).

Réponses aux quiz :

	A	B	C	D	
question 1 :				✓	1
question 2 :			✓		2
question 3 :		✓			3
question 4 :	✓				4
question 5 :			✓		5
question 6 :			✓		6

	A	B	C	D	
question 7 :		✓			7
question 8 :			✓		8
question 9 :			✓		9
question 10 :				✓	10
question 11 :				✓	11
question 12 :			✓		12

PARTIE QUIZ

1 – Utilisation d'un ordinateur [2 points]

Question 1) Sur un ordinateur avec un processeur à 32 bits, combien de temps faut-il pour lire (en mémoire) un fichier de 12 Mo si le disque transfère 6000 *mots* par ms ?

A] 0.25 s

B] 0.5 μ s

C] 2.5 s

✓D] 0.5 s

Question 2) Considérons deux ordinateurs O_1 et O_2 de fréquences de processeur respectives 3.2 et 2 GHz, de tailles de cache respectives 6 et 2 Mo et ayant respectivement un disque dur de bande passante 30 et 60 Mo/s ; et considérons trois programmes :

- P_1 , qui utilise beaucoup de mémoire et peu de disque ;
- P_2 , qui utilise beaucoup le disque ;
- et P_3 , qui utilise beaucoup le CPU et moins de 1 Mo de mémoire.

Lesquelles des comparaisons suivantes sont toutes les trois correctes si l'on compare les vitesses d'exécution des programmes ?

A] $P_1 : O_1 > O_2$; $P_2 : O_1 > O_2$; $P_3 : O_1 < O_2$

B] $P_1 : O_1 < O_2$; $P_2 : O_1 < O_2$; $P_3 : O_1 > O_2$

✓C] $P_1 : O_1 > O_2$; $P_2 : O_1 < O_2$; $P_3 : O_1 > O_2$

D] $P_1 : O_1 < O_2$; $P_2 : O_1 > O_2$; $P_3 : O_1 < O_2$

2 – Caches [4 points]

Question 3) Sur un ordinateur avec une mémoire cache, on calcule le vecteur Y résultat du produit d'une matrice M par un vecteur X . Laquelle des propositions suivante est vraie ?

A] Si le cache est assez grand pour contenir M , X et Y en même temps, alors il n'y aura aucun défaut de cache.

✓B] Si la mémoire est organisée et le cache assez grand de sorte à pouvoir contenir X , Y et une ligne de M en même temps, alors le nombre de défauts de cache est le même que si le cache était assez grand pour contenir M , X et Y en même temps.

C] Si la mémoire est organisée et le cache assez grand de sorte à pouvoir contenir X , Y et une colonne de M en même temps, alors le nombre de défauts de cache est plus petit que si le cache était assez grand pour contenir M , X et Y en même temps.

D] Si le cache est assez grand pour contenir X , Y et une colonne de M en même temps, alors le nombre de défauts de cache ne dépend pas de la façon dont la matrice M est stockée en mémoire.

Question 4) Laquelle des propositions suivantes est vraie ?

✓A] Changer l'ordre de boucles dans un programme peut améliorer la localité temporelle.

B] Plus les blocs mémoire sont grands, plus on a de localité temporelle.

C] La localité spatiale des variables d'un programme dépend de la taille du cache.

D] Le nombre de défauts de cache ne dépend pas de la localité spatiale.

suite au dos ➡

Question 5) La politique de remplacement de la mémoire cache vue en cours (LRU) :

- A] minimise le nombre de remplacements de blocs.
- B] maximise le nombre de remplacements de blocs.
- ✓C] remplace le bloc utilisé il y a le plus longtemps.
- D] remplace le bloc utilisé le moins souvent.

Question 6) En utilisant la politique de remplacement de cache LRU vue en cours, et sachant que la mémoire cache peut contenir 2 blocs de 4 mots, combien de défauts de cache génère la séquence d'accès mémoire suivante (adresses) :

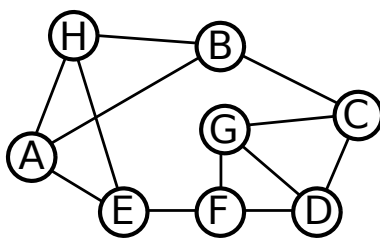
22 21 14 15 9 20 8 5 4 3 10 8 1 2

(L'adressage mémoire se fait par mot et les blocs sont toujours alignés sur les multiples de 4.)

- A] 5 B] 6 ✓C] 7 D] 8

3 – Routes, encore [2 points]

Question 7) Laquelle des lignes suivantes fait partie de la table de routage de A pour ce réseau :



- A]

dest.	dir.	dist.
C	E	4

 ✓B]

dest.	dir.	dist.
C	B	2

 C]

dest.	dir.	dist.
E	H	2

 D]

dest.	dir.	dist.
E	F	2

Question 8) Le réseau de PolyLAN possède un routeur principal qui a l'adresse IPv4 interne 10.0.2.1 sur son interface **eth0** (relié au réseau des joueurs) et l'adresse IP externe 128.179.99.2 sur son interface **eth1**. L'interface **eth1** est reliée à un sous-réseau de l'EPFL sur lequel un routeur à l'adresse 128.179.99.1 (sur son interface **eth1**) s'occupe du trafic vers le réseau de SWITCH (130.59.36.0). SWITCH transfère ensuite le trafic vers Internet au travers d'un routeur ayant l'IP 130.59.36.1.

A l'intérieur du réseau de PolyLAN se trouve un routeur secondaire à l'adresse 10.0.2.2 sur son interface **eth1** (relié au réseau des joueurs) et 10.0.3.1 sur son interface **eth0** (relié au réseau des serveurs, 10.0.3.0).

Laquelle de ces entrées ne peut pas figurer dans la table de routage du routeur *secondaire* de PolyLAN :

- A]

Destination	Passerelle	Interface	Métrique
10.0.2.0	0.0.0.0	eth1	0
- B]

Destination	Passerelle	Interface	Métrique
128.179.99.0	10.0.2.1	eth1	1
- ✓C]

Destination	Passerelle	Interface	Métrique
130.59.36.0	128.179.99.1	eth0	2
- D]

Destination	Passerelle	Interface	Métrique
10.0.3.0	0.0.0.0	eth0	0

4 – TCP [1 point]

Question 9) Deux ordinateurs communiquent par TCP/IP en passant par un routeur. Supposons qu'un unique paquet IP de l'ordinateur 1 est perdu sur le lien routeur—ordinateur 2.

Le paquet en question

- A] n'est pas renvoyé car TCP ne garantit pas la bonne réception des informations.
- B] est renvoyé par le routeur.
- ✓C] est renvoyé par l'ordinateur 1.
- D] est reconstruit à l'aide de codes de Huffman.

5 – Couches de protocole [1 point]

Question 10) Au bureau, mon ordinateur se connecte au Web (HTTP) par le wifi et envoie ses emails (SMTP) par câble Ethernet (câble de cuivre). A quels deux niveaux ces deux communications sont-elles le plus fondamentalement différentes ?

- | | |
|------------------------------|------------------------------|
| A] Application et Transport. | C] Transport et Réseau. |
| B] Réseau et Physique. | ✓D] Application et Physique. |

6 – Assembleur [1 point]

Question 11) En supposant x , y et z , entiers positifs, stockés au départ respectivement dans `r0`, `r1` et `r2`, que calcule le code assembleur suivant, où l'instruction « `cont_poseq a, b` » saute à la ligne `b` si la valeur `a` est positive ou nulle :

```
1: somme r3, r0, r1
2: charge r4, -1
3: somme r3, r3, -r2
4: somme r4, r4, 1
5: cont_poseq r3, 3
```

- A] $(x + y) \times z$ B] $x + y - z$ C] $(x + y - 1) \times (-z)$ ✓D] $(x + y)/z$ (division entière)

7 – Circuits [1 point]

Question 12) Quelle est la table de vérité de la sortie `X` du circuit ci-contre ?

Aucune, ce circuit
A] peut causer des
courts-circuits.

C] ✓

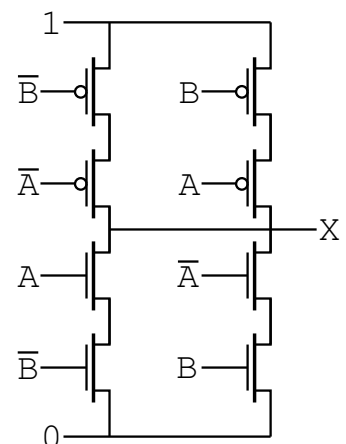
		A	
		0	1
B	0	1	0
	1	0	1

B]

		A	
		0	1
B	0	0	1
	1	1	0

D]

		A	
		0	1
B	0	1	1
	1	0	1



PARTIE EXERCICES

8 – Alignement de séquences [4 points]

Soit C une matrice de taille $n \times n$. On veut calculer les coefficients d'une matrice M de taille $(n+1) \times (n+1)$ telle que :

- $M(0, j) = 0$ pour tout $j, 0 \leq j \leq n$;
- $M(i, 0) = 0$ pour tout $i, 0 \leq i \leq n$;
- $M(i, j) = C(i, j) + \max(M(i, j-1), M(i-1, j), M(i-1, j-1))$ pour tout les autres indices.

1. (1 point) Ecrire un algorithme pour calculer M .
2. (0.5 point) Quelle est la complexité de votre algorithme ?
3. (2.5 points) On suppose que les matrices sont stockées par lignes dans la mémoire, laquelle a des blocs de 4 mots de 64 bits : C est stockée à partir de l'adresse 4 et M à partir de l'adresse 68. Les valeurs des matrices sont codées sur 64 bits. n vaut 8 et est stocké (sur 64 bits) à l'adresse 2. On suppose de plus que l'on a un cache (LRU) de 8 blocs et que l'utilisation de i, j et n est optimisée par le compilateur via des registres.

Combien de défauts de cache se produisent lors du calcul de la ligne $M(2, \cdot)$? Justifiez votre réponse.

Il est important que les cases $M(i, j-1)$, $M(i-1, j)$ et $M(i-1, j-1)$ aient été calculées avant la case $M(i, j)$. Il y a pour cela plusieurs parcours possibles, en voici un :

alignement
entrée : C matrice de taille $n \times n$ sortie : M matrice du meilleur alignement
Pour i de 0 à n $M(0, i) \leftarrow 0$ $M(i, 0) \leftarrow 0$ Pour i de 1 à n Pour j de 1 à n $M(i, j) \leftarrow C(i, j) + \max(M(i, j-1), M(i-1, j), M(i-1, j-1))$ sortir : M

Il est en particulier important que l'initialisation ait été faite *avant* le parcours principal.

L'algorithme ci-dessus est en $\mathcal{O}(n^2)$.

Etat de la mémoire :

0 | - | - | n | - || 4 $C_{1,1}$ | $C_{1,2}$ | 3 | 4 || 8 5 | 6 | 7 | 8 || 12 $C_{2,1}$ | $C_{2,2}$ | 3 | 4 || 16 5 | 6 | 7 | 8 || ... |
68 $M_{0,0}$ | 1 | 2 | 3 || 72 4 | 5 | 6 | 7 || 76 8 | $M_{1,0}$ | 1 | 2 || 80 3 | 4 | 5 | 6 || 84 7 | 8 | $M_{2,0}$ | 1 || 88 2 | 3 | 4 | 5 || 92 6 | 7 | 8 |

État du cache après le calcul de $M(1, \cdot)$ (plus récent à droite, alternatives possibles par lignes) :

0 4 68 80 72 76 8 84
 68 4 76 72 8
 8 72 76
 8 76 72

Calcul de $M(2, \cdot)$, utilisation des blocs :

84

76 ou 12 MISS 12 --> 0

l'autre (76 ou 12)
88 MISS 88 --> 4 ou 68
84 ou 76 ou 12
les 2 autres
etc.
80 (remonte)
16 MISS 16 --> 4 ou 68
etc.
92 MISS 92 --> 8 ou 72

\Rightarrow 4 défauts de cache en tout.

Barème :

1. 0.5 pour l'initialisation ; 0.5 pour le calcul
2. aucun point si la complexité ne correspond pas à celle de l'algorithme donné (donc en particulier 0 si pas d'algorithme !)
3. 1 pt pour penser à la réutilisation de $M(1,)$; 0.5 pt pour voir le décalage de $M(2,)$ en mémoire ; 1 pt pour les principes du compte lui-même : réutilisation du passé, défaut de cache pour ce qui est nouveau, ...

Si la réponse est fournie juste mais sans aucune justification : 0.5 pt (bénéfice du doute). Si par contre elle est fournie avec des justifications erronées : 0 pt (il y a plein de mauvaises raisons de trouver un « 4 »).

9 – Réchauffement [4 points]

Le petit Néo veut créer un système de relevé automatique de la température extérieure. Il sait que son thermomètre électronique a une précision de 0.5°C et que chaque mesure sera stockée avec une indication du temps à laquelle elle aura été prise, exprimé en secondes depuis le 1^{er} janvier 1970 (appelé « temps Unix »).

Le tout sera stocké dans une table d'une base de données avec les deux colonnes mentionnées : le temps comme un entier sur 32 bits et la température comme un nombre à virgule flottante sur 32 bits avec 23 bits de mantisse.

1. (1 point) Tout les combien de temps (au minimum) est-ce que son système doit sauvegarder une mesure de température si Néo veut pouvoir reconstruire exactement la courbe des températures et que l'on suppose qu'elles ne varient pas à plus de 0.0015 Hz.

Réponse : échantillonner à au moins $2f$, soit au moins toutes les 333 secondes.

2. (1 point) Sachant que Néo a commencé à enregistrer ses températures à partir du temps Unix écrit en binaire 0101 0100 1001 0100 0011 0010 1110 0100, à quoi correspond l'enregistrement écrit en binaire sur le disque comme :

0101 0100 1001 0100 0011 0110 0110 1000, 0000 0001 1000 1000 0000 0000 0000 0000

(date en premier, puis signe, exposant – ici directement en binaire – et mantisse de la température)? Quelle est la température?

Expliquez votre réponse.

Réponse : 15 min (= 900 sec.) après le début de ses mesures, la température était de 8.5°C

3. (2 points) Comment pourrait faire Néo pour réduire la taille des informations stockées, sans perdre d'information? Essayez de lui proposer un système optimal.

Réponse : s'il est sûr de prendre ses mesures à intervalles réguliers, sans en perdre, alors rien ne sert de stocker le temps. Sinon, on peut stocker le temps à partir de 0 au lieu de 1970 et gagner plusieurs bits sur la date : par exemple pour 10 ans si l'on mesure toutes les 5 minutes, il suffit de 20 bits.

Pour les températures : entre -50 et + 50 semble largement raisonnable, à raison de 0.5 cela fait 200 mesures possibles, soient 8 bits.

Ensuite, on pourrait étudier l'entropie des températures et certainement y gagner encore (car elles sont loin d'être équiprobables) avec un code de Huffman. On pourrait aussi envisager de coder les écarts entre valeurs successives au lieu des valeurs elles-mêmes.

Barème :

1. 0.5 pour $2f$; 0.25 pour la valeur (de $2f$) ; 0.25 pour la valeur en temps (dont 0.125 pour « temps = inverse fréquence »), mais 0 si on n'a pas le th. (i.e. si on n'a pas $2f$).
2. 0.25 pour le principe du calcul de la date (différence), le principe suffit, on n'attend pas ici le calcul complet ; 0.25 pour le calcul de l'exposant ; 0.25 pour la mantisse complète (1.m ; que 0.125 s'il manque le 1) ; 0.25 pour la valeur (que 0.125 si le principe « exposant fois 1.m » y est, mais erreur de calcul).

S'il n'y a que le signe, expliqué, et rien d'autre, mettre 0.125.

3. 0.5 pour coder le temps sur moins de bits ; 0.5 pour coder la température sur moins de bits (ensemble fini de valeurs, lié à la précision de mesure) ; 0.75 pour mener la réflexion plus loin sur les codes compresseurs (sans perte), dont 0.25 pour citer Huffman ; et enfin 0.25 pour dire que l'entropie des températures est certainement faible (et donc qu'on peut compresser).

Pour les 0.75 de « compression », on peut *par exemple* donner 0.25 pour parler de compression (sans perte), 0.25 pour parler de code (vu en cours/exercices) et 0.25 pour l'optimalité/Huffman.