UNIVERSITÉ DE SHERBROOKE
DÉPARTEMENT D'INFORMATIQUE

IFT 436
Algorithmes et structures de données
EXAMEN PÉRIODIQUE[1]

Le mercredi 15 juin 2016 de 10 h 30 à 12 h 20   Professeur : Richard St-Denis

- Toute documentation est permise.
- **Tout appareil électronique, incluant le téléphone cellulaire et l'ordinateur, est interdit.**
- Ne dégrafez pas ce questionnaire.
- Répondez dans les espaces prévus à cet effet.
- **Personne ne peut quitter la salle d'examen avant 10 h 45.**
- **Personne ne peut quitter son siège entre 12 h 10 et 12 h 20.**
- La correction est, entre autres, basée sur le fait que chacune de vos réponses soit :
  - claire, c'est-à-dire lisible et compréhensible pour le correcteur ;
  - précise, c'est-à-dire exacte ou sans erreur ;
  - complète, c'est-à-dire que toutes les étapes de résolution du problème sont présentes ;
  - concise, c'est-à-dire que la méthode de résolution est la plus courte possible.

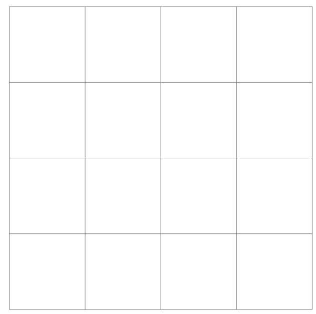
Nom : ______________________ Prénom : ______________________

Signature : ______________________ Matricule : ______________________

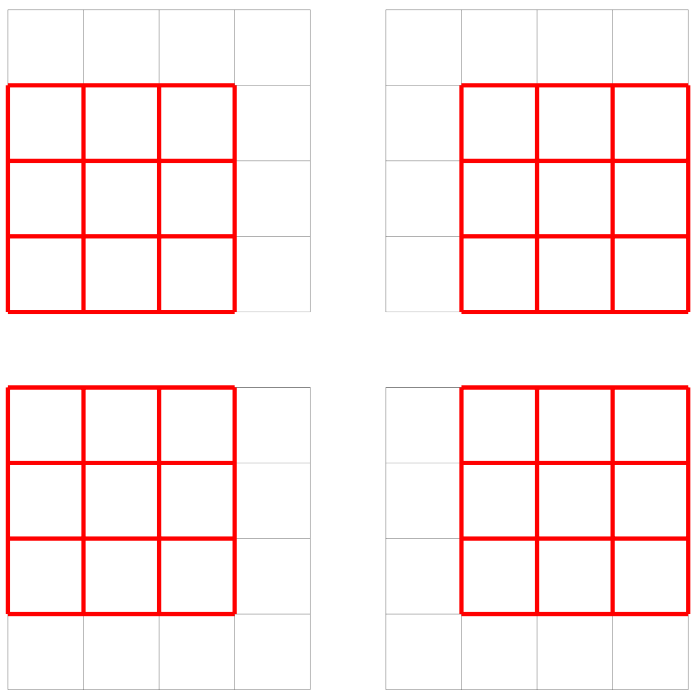
Question		Barème
1		/7
2		/8
3		/7
4		/8
	total :	/30

1. Ce questionnaire comporte huit (8) pages.

1. (7 points) On considère une grille de n par n cases, $n \geq 3$. Par exemple voici une grille 4×4 :



Vous devez déterminer le nombre de sous-grilles 3×3 dans une grille $n \times n$. Par exemple, il y a quatre sous-grilles 3×3 dans une grille 4×4 :



a) (2 points) Exprimez, à l'aide d'une équation de récurrence, le nombre de sous-grilles 3×3 dans une grille $n \times n$ Donnez aussi la condition initiale. Indiquez brièvement le raisonnement que vous avez suivi.

b) (2 points) Devinez une solution (une formule sous la forme fermée) pour le nombre de sous-grilles 3×3 dans une grille $n \times n$. Expliquez informellement votre solution par un raisonnement différent de celui utilisé en a).

c) (3 points) Montrez par induction que la solution obtenue en b) est bien une solution de l'équation de récurrence obtenue en a).

2. (8 points) Soit $f : \{0, 1, \ldots 2^n - 1\} \to \{0, 1\}$ une fonction binaire constante ou équilibrée. Rappelons qu'une fonction est constante si $f(x) = 0$ ou $f(x) = 1$ pour tout $x \in \{0, 1, \ldots 2^n - 1\}$. Elle est équilibrée si $|f^{-1}(0)| = 2^{n-1}$, où $f^{-1}(y) = \{x \in \{0, 1, \ldots 2^n - 1\} \mid f(x) = y\}$ pour $y \in \{0, 1\}$.

   Voici les deux algorithmes présentés en classe, sous la forme de pseudo-code, qui détermine si f est équilibrée ou constante.

   Premier algorithme :

$f_0 := f(0)$
$\{$ each i : $[1..2^{n-1}]$ $|$
    $f_i := f(i)$
    if $f_{i-1} \neq f_i$ then ``balanced'' exit $\}$
``constant''

   Deuxième algorithme :

$\{$ each i : $[0..2^{n-1}]$ $|$ $f_i^0 := f(i)$ $\}$
$\{$ each k : $[1..n-1]$ $|$ // $n - 1 = \log 2^{n-1}$
    $\{$ each i : $[0..\frac{2^{n-1}}{2^k} - 1]$ $|$
        if $f_{2i}^{k-1} \neq f_{2i+1}^{k-1}$ then ``balanced'' exit
        $f_i^k := f_{2i}^{k-1}$ $\}$ $\}$
if $f_0^{n-1} \neq f_{2^{n-1}}^0$ then ``balanced'' else ``constant''

   a) (1 point) Indiquez le nombre **exact** d'affectations (« := ») effectuées par le premier algorithme dans le meilleur cas et dans le pire cas. Justifiez vos réponses.

   Meilleur cas : ______

   Pire cas : ______

   b) (1 point) Indiquez le nombre **exact** de comparaisons (« $\neq$ ») effectuées par le premier algorithme dans le meilleur cas et dans le pire cas. Justifiez vos réponses.

   Meilleur cas : ______

   Pire cas : ______

c) (2 points) Indiquez le nombre **exact** d'affectations (« := ») effectuées par le deuxième algorithme dans le meilleur cas et dans le pire cas. Justifiez vos réponses.

Meilleur cas : ______________________________

____

____

Pire cas : ______________________________

____

____

____

____

____

____

d) (2 points) Indiquez le nombre **exact** de comparaisons (« $\neq$ ») effectuées par le deuxième algorithme dans le meilleur cas et dans le pire cas. Justifiez vos réponses.

Meilleur cas : ______________________________

____

____

____

Pire cas : ______________________________

____

____

____

____

____

e) (1 point) Que pouvez-vous conclure sur la performance des algorithmes, l'un par rapport à l'autre ?

____

____

____

f) (1 point) Que pouvez-vous conclure sur la complexité de calcul de chacun des algorithmes exprimée dans les notations asymptotiques ?

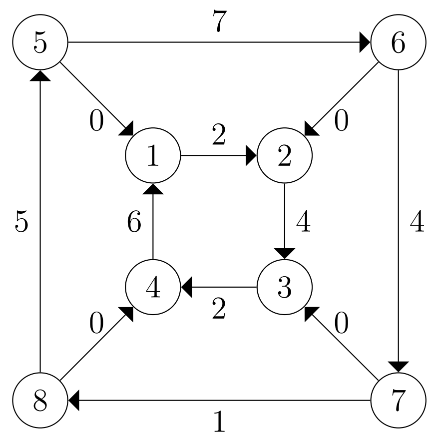
____

____

____

____

3. (7 points) Voici un graphe orienté valué.



a) (3 points) Illustrez l'exécution de l'algorithme DFS (recherche en profondeur) à partir du graphe G. Considérez que les sommets sont examinés selon l'ordre croissant (du plus petit numéro au plus grand numéro) et que les sommets adjacents à un sommet apparaissent aussi en ordre croissant dans la liste d'adjacence. Donnez la solution sous la forme d'une forêt d'arborescences avec les estampilles de début et de fin, respectivement à gauche et à droite de chaque sommet.

b) (4 points) Illustrez l'exécution de l'algorithme de Kruskal (calcul de l'arbre sous-tendant de coût minimal) à partir de la version non orientée du graphe G.

4. (8 points) Une *clique* d'un graphe non orienté $G = (V, E)$ est un sous-ensemble des sommets de V qui induit un sous-graphe complet, c'est-à-dire que deux sommets quelconques de la clique sont adjacents.

   a) (3 points) Concevez un algorithme, présenté sous la forme de **pseudocode**, qui détermine si le sous-graphe induit par $V' \subseteq V$ est une clique lorsque le graphe G est représenté par une matrice d'adjacence.

   b) (3 points) Concevez un algorithme, présenté sous la forme de **pseudocode**, qui détermine si le sous-graphe induit par $V' \subseteq V$ est une clique lorsque le graphe G est représenté par des listes d'adjacence.

   c) (2 points) En utilisant la stratégie par la force brute, concevez un algorithme, présenté sous la forme de **pseudocode**, qui décide s'il existe une clique, la plus grande possible, induite par un sous-ensemble V' de V, avec la cardinalité de V' égale à k, $2 \leq k \leq v(G)$ (i.e., $|V'| = k$).

FIN DE L'EXAMEN