21 avril 2004

UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS
DÉPARTEMENT D'INFORMATIQUE

STRUCTURES DES INFORMATIONS I
INF4063

EXAMEN FINAL

Directives:

1. C'est un examen à livre ouvert.
2. Répondre directement sur ce formulaire.
3. Les questions éventuelles doivent être posées durant les 30 premières minutes.
4. L'examen dure trois heures.
5. Le barème est montré au début de chaque question.

Nom, prénom:____

Code permanent:____

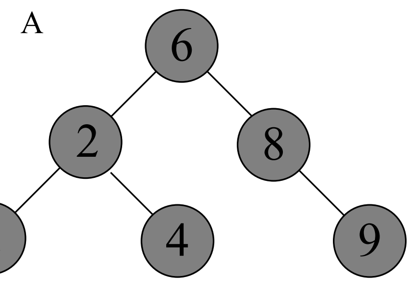
Signature:____

Numéro de formulaire:_______________________________________

Ex. #	1	2	3	4	5	6	7	8	9	10	11	Σ
Note												
sur	9	9	10	8	10	9	9	10	7	10	9	100

Ex. #1 (9 pts)

Soit A un arbre binaire de recherches suivant:



a) (1 pts) Supposons que l'on a performé la suppression de la clé 8 de l'arbre A. Donner la forme de l'arbre après cette suppression:

b) (1 pts) Supposons que l'on a performé l'insertion de la clé 7 dans l'arbre A (tel que donné initialement). Donner la forme de l'arbre après cette insertion:

c) (2 pts) Supposons que l'on a performé la suppression de la clé 2 de l'arbre A (tel que donné initialement). Donner la forme de l'arbre après cette suppression:

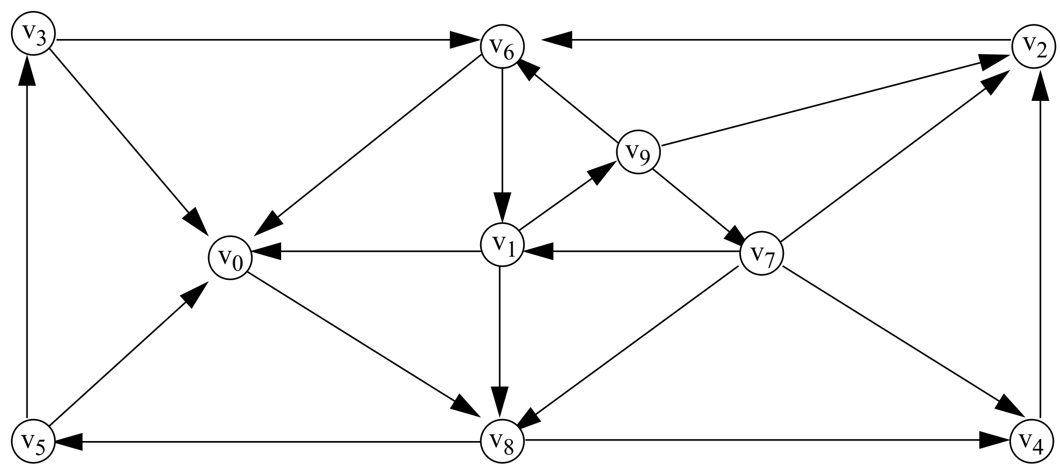
d) (5 pts) Supposons que pour l'arbre A (tel que donné initialement) on a performé une séquence de requêtes suivantes:

Ajouter (7);
Ajouter (5);
Supprimer (6);
Ajouter (6);
Supprimer (7);

Donner une suite de 5 arbres qui sont les résultats après chacune de ces requêtes (l'objet de chaque requête est l'arbre obtenu à la suite des requêtes précédentes).

Ex. #2 (9 pts)

Soit donné un graphe dirigé G suivant:

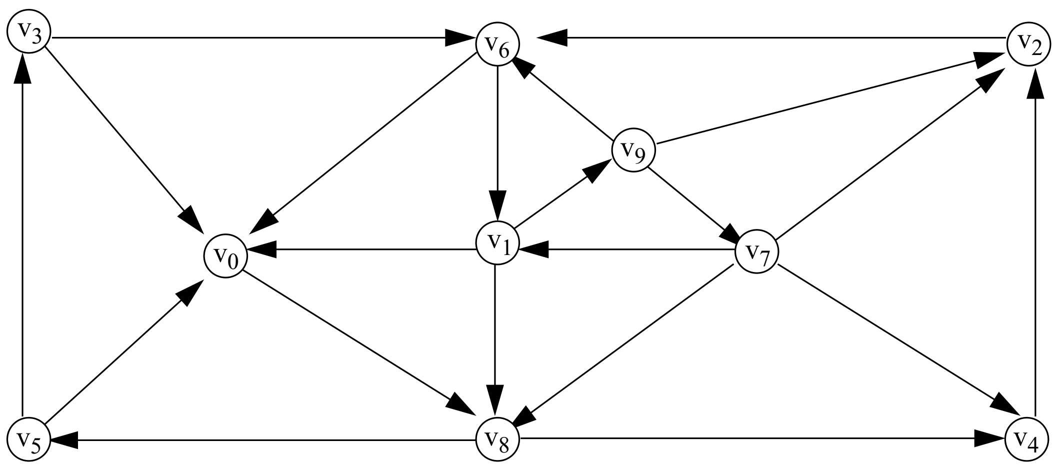


Pour le graphe G on a exécuté l'algorithme du plus court chemin à partir du sommet source v_0 - PLUS-COURT-SANS-POIDS (G, v_0) donné ci-dessous. On suppose que la boucle **pour** de la ligne 6 considère les sommets adjacents à u dans l'ordre croissant des numéros de ces sommets.

```
PLUS-COURT-SANS-POIDS (G, s)
1       INITIALISER (G, s)
2       file F ← vide
3       ENFILER (F, s)
4       tant que F ≠ vide
5               u ← DEFILER (F)
6               pour chaque sommet v adjacent à u faire
7                       si distance (v) = ∞ alors
8                               distance (v) ← distance (u) + 1
9                               PARENT (v) ← u
10                              ENFILER (F, v)
11              afficher la file F
```

```
INITIALISER (G, s)
1       pour chaque sommet s de G
2               faire   distance (s) ← ∞
3                       PARENT (u) ← NIL
4       distance (s) ← 0
```

a) (4.5 pts) À partir du graphe G sur l'image ci-dessous choisissez les arêtes qui forment l'arbre des distances les plus courtes créé par cet algorithme (à la ligne 9). Sur chaque sommet marquez la distance de ce sommet calculée par l'algorithme.



b) (4.5 pts) Donner la suite d'opérations ENFILER (F, u) et u ← DEFILER (F) qui se performe durant l'exécution de cet algorithme:

Ex. #3 (10 pts)

Dans un tas-max TM il se trouve sept clés différents ayant comme valeurs: 1, 2, 3, 4, 5, 6, et 7.

a) (1.5 pts) Mettez les valeurs de 1 à 7 dans un tableau TM pour que l´état de ce tableau représente un tas-max valide.

TM

1	2	3	4	5	6	7

b) (1.5 pts) Supposons que le tableau TM est tel que TM[5]=5 et TM[6]=3. Mettez les autres valeurs dans letableau TM pour que l´état de ce tableau représente un tas-max valide.

TM

1	2	3	4	5	6	7
				5	**3**	

c) (1.5 pts) De combiens façons on peut completer le tableau TM du point b) ci-dessus pour que l´état de ce tableau représente un tas-max valide?

d) (2 pts) De combiens façons on peut completer le tableau TM donné ci-dessous pour que l´état de ce tableau représente un tas-max valide?

TM

1	2	3	4	5	6	7
				5		

e) (3.5 pts) De combiens façons on peut completer le tableau TM du point a) pour que l´état de ce tableau représente un tas-max valide? Autrement dit, combiens de tas-max différents peut-on construire à partir d'un ensemble de sept clés?

Ex. #4 (8 pts)

L'arbre A de l'exercise #1 est un arbre binaire de recherche de profondeur deux (chaque sommet de A a une distance d'au plus deux à partir de la racine).

a) (1.5 pts) Combiens d'arbres binaires de recherche de profondeur deux on peut construire à partir de sept clés différents?

b) (2.5 pts) Combiens d'arbres binaires de recherche de profondeur deux on peut construire à partir de six clés différents?

c) (4 pts) Combiens d'arbres binaires de recherche de profondeur deux on peut construire à partir de cinq clés différents?

Ex. #5 (10 pts)

Soit LB une liste entière bidirectionelle. Donner une fonction qui supprime de LB tous les élements avec des valeurs paires.

Ex. #6 (9 pts)

Donner la forme préfixée pour chacune des expressions postfixées suivantes.

a) a b c d / - *

b) a b c + e f >= / =   /* opérateurs du langage C */

c) a b -- c - d e != f * >   /* opérateurs du langage C, y compris l'opérateur unaire --*/

Ex. #7 (9 pts)

Donner la forme infixée pour chacune des expressions de l'ex. #2, en mettant juste les parenthèses absolument nécessaires

a)

b)

c)

Ex. #8 (10 pts)

Soit une expression de langage C suivante:

```
a+b*c>d->e|(f-g*h)<i+j%k.l
```

a) (4.5 pts) Donner la forme postfixée de cette expression

b) (5.5 pts) Donner une suite d'actions qui se deroulent sur la pile (Empiler et Dépiler) durant la transformation de cette expression infixée à sa forme postfixée

Ex. #9 (7 pts)

Donner l'arbre de l'expression donnée dans l'ex #8.

Ex. #10 (10 pts)

a) (3 pts) Voici l'état du tableau TM représentant un tas-max contenant un ensemble de douze clés:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
30	26	24	10	16	18	4	2	8	6	12	14			

Quel sera l'état du tableau TM après avoir performé la suppression de la clé maximale de TM?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

b) (7 pts) Voici l'état du tableau TM qui ne représente pas encore un tas-max contenant un ensemble de douze clés:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	4	6	2	18	8	10	12	30	20	22	14			

Quel sera l'état du tableau TM après avoir performé l'opération CONSTRUIRE-TAS-MAX(TM)?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Ex. #11 (9 pts)

Soit une fonction récursive pui donnée ci-dessus:

```
int pui (int x, unsigned int n)
{
    if (n == 0)
            return 1;
    if (n == 1)
            return x;
    if (n % 2 == 0)                                       /* n est pair */
            return ( pui (x*x, n/2) );
    else                                                  /* n est impair */
            return (pui (x*x, n/2) * x );
}
```

Donner la valeur retournée par l'appel de la fonction

a) (1 pt) pui (3, 0)

b) (1 pt) pui (5, 1)

c) (1.5 pts) pui (2, 2)

d) (2 pts) pui (3, 3)

e) (3.5 pts) pui (p, m) /* donner cette valeur en fonction de paramètres p et m */