

## Questionnaire examen final


**INF2010**

Sigle du cours

<i>Identification de l'étudiant(e)</i>		
Nom :	Prénom :	
Signature :	Matricule :	Groupe :

<i>Sigle et titre du cours</i>		<i>Groupe</i>	<i>Trimestre</i>
INF2010 – Structures de données et algorithmes		Tous	20093
<i>Professeur</i>		<i>Local</i>	<i>Téléphone</i>
Thierry M. Lavoie et Tarek Ould Bachir (chargés de cours) Ettore Merlo (coordonateur)		M-4105	4193 / 5193
<i>Jour</i>	<i>Date</i>	<i>Durée</i>	<i>Heures</i>
Lundi	07 décembre 2009	2h30	13h30-16h00

<i>Documentation</i>	<i>Calculatrice</i>	
<input checked="" type="checkbox"/> Aucune <input type="checkbox"/> Toute <input checked="" type="checkbox"/> Voir directives particulières	<input type="checkbox"/> Aucune <input type="checkbox"/> Toutes <input checked="" type="checkbox"/> Non programmable	Les cellulaires, agendas électroniques ou téléavertisseurs sont interdits.

<i>Directives particulières</i>
<p> Un cahier supplémentaire vous sera remis. Servez-vous de ce cahier comme brouillon. Nous ne récupérerons pas le cahier supplémentaire à la fin de l'examen.</p> <p style="text-align: right;"><i>Bonne chance à tous!</i></p>

<b>Important</b>	<p>Cet examen contient <b>8</b> questions sur un total de <b>13</b> pages (incluant cette page)</p> <p>La pondération de cet examen est de <b>40</b> %</p> <p>Vous devez répondre sur : <input checked="" type="checkbox"/> le questionnaire <input type="checkbox"/> le cahier <input type="checkbox"/> les deux</p> <p>Vous devez remettre le questionnaire : <input checked="" type="checkbox"/> oui <input type="checkbox"/> non</p>
------------------	--

**L'étudiant doit honorer l'engagement pris lors de la signature du code de conduite.**



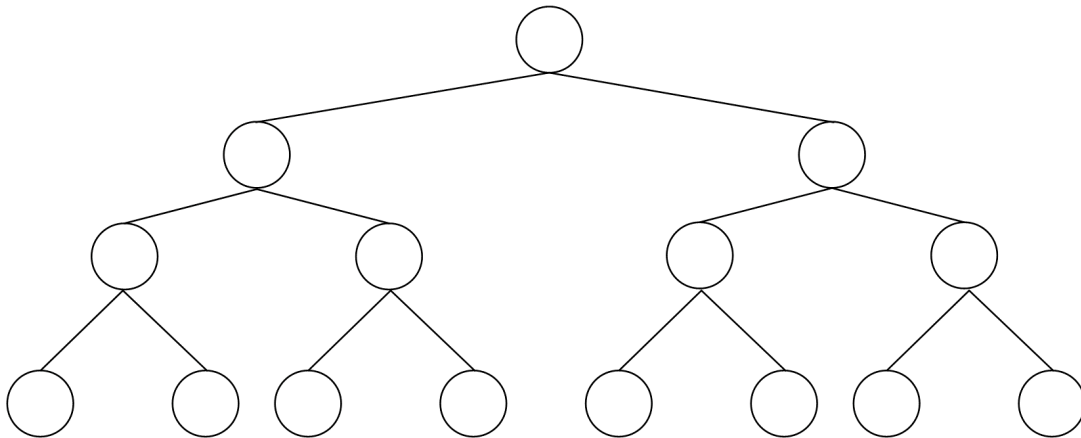
**Question 2 : Tri par monceau****(15 points)**

On désire trouver les trois plus petits éléments d'un vecteur d'entiers. On vous suggère, pour ce faire, d'utiliser un monceau MIN et de trouver les trois plus petits éléments au niveau de la racine et de ses deux enfants.

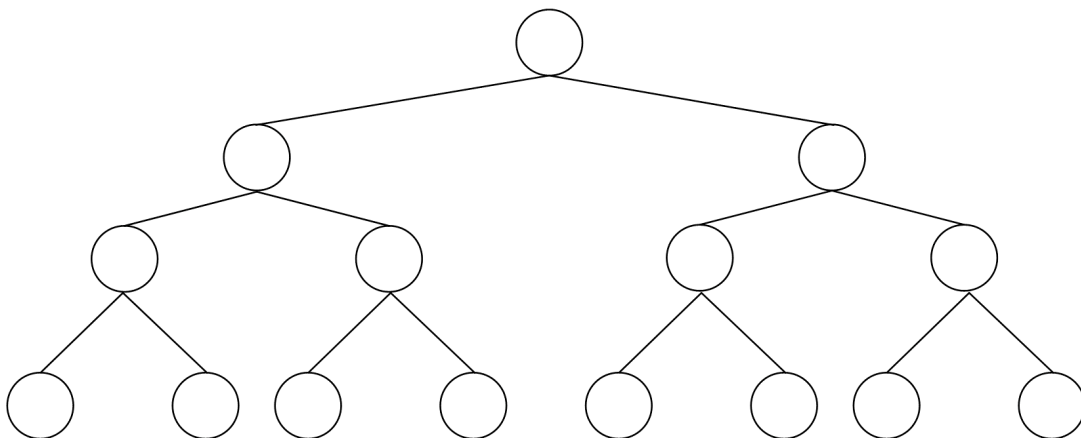
2.1) (3 points) Commentez cette stratégie. Est-elle justifiée ou non ? Expliquez clairement votre réponse.

2.2) Illustrez votre réponse en construisant un monceau min de chacun des vecteurs suivants :

2.2.1) (1.5 point) 15, 31, 21, 10, 20, 14, 5, 12, 2, 4, 1, 3, 7, 19, 35

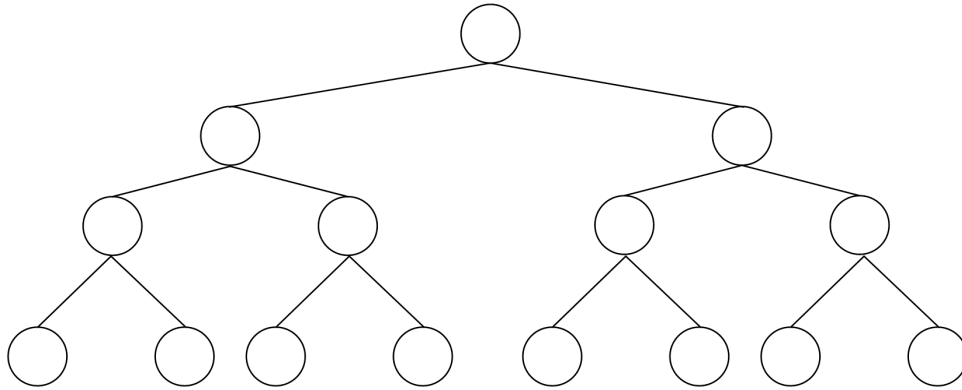


2.2.2) (1.5 point) 15, 31, 21, 10, 20, 14, 5, 12, 2, 3, 1, 4, 7, 19, 35

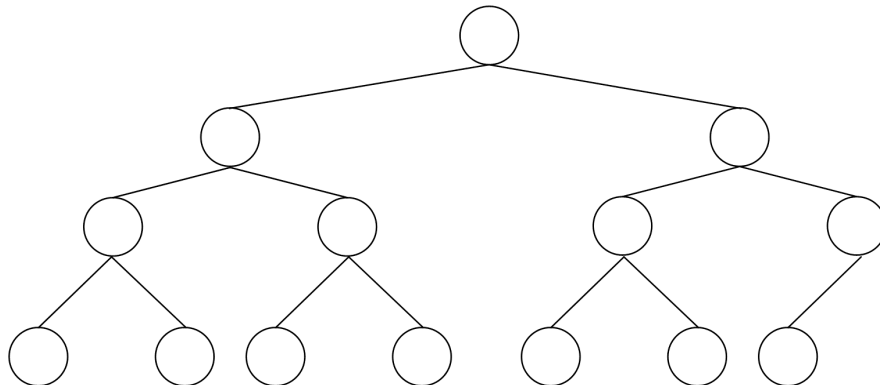


2.3) Exécutez l'algorithme de tri partiel vu en cours pour trouver les trois plus petites valeurs parmi les entiers 15, 31, 21, 10, 20, 14, 5, 12, 2, 3, 1, 4, 7, 19, 35 (ceux de 2.2.2). Pour mémoire, le principe du tri partiel consiste à construire un monceau MIN et à retirer la racine pour chaque nouvelle valeur.

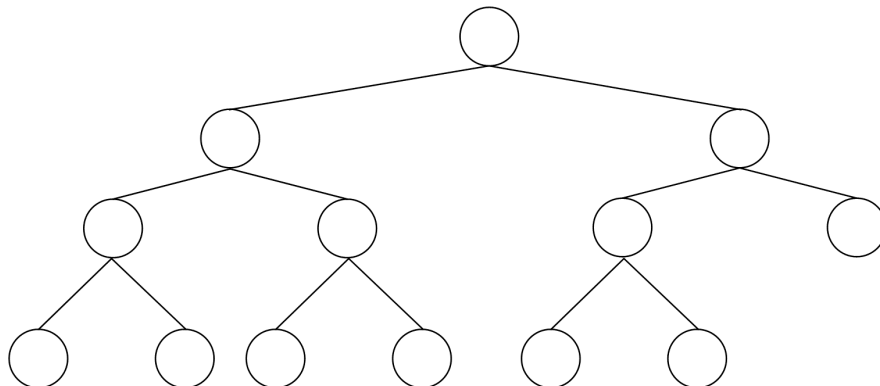
2.3.1) (3 points) Premier monceau du tri partiel, avant retrait de la racine :



2.3.2) (3 points) Second monceau du tri partiel, avant retrait de la racine :



2.3.3) (3 points) Troisième monceau du tri partiel, avant retrait de la racine :

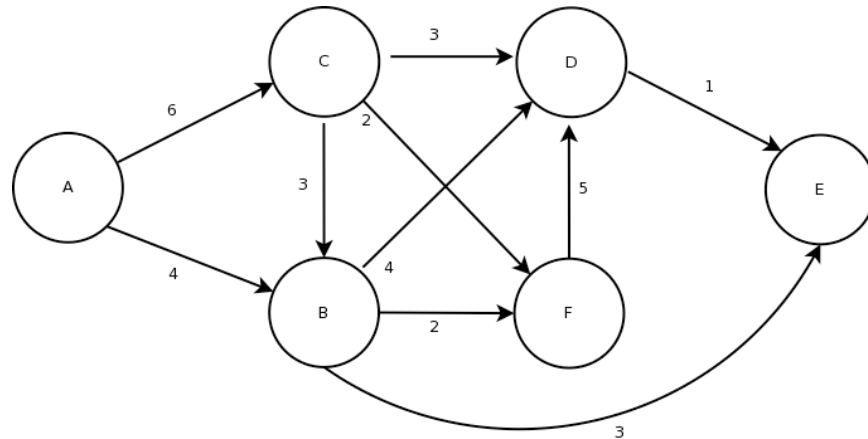


**Question 3 : Modélisation par graphes****(15 points)**

- 3.1) (5 points) Soit un labyrinthe composés d'un nombre  $N$  de salles. Chaque salle est de forme carrée et peut avoir jusqu'à 4 portes conduisant à une salle adjacente, soit au nord, au sud, à l'est ou à l'ouest. Proposez une façon de modéliser ce problème avec un graphe. Indiquez clairement ce que représente les nœuds et les arrêtes.
- 3.2) (5 points) Supposons que l'on vous donne un graphe  $G(V, E)$  qui respecte votre modélisation de la question 3.1). Donnez, en notation asymptotique, le temps d'exécution le plus rapide qu'il est possible d'obtenir pour le problème consistant à trouver la sortie du labyrinthe à partir du point de départ.
- 3.3) (5 points) Supposons maintenant que l'on permette de détruire des murs pour trouver un chemin, mais la destruction entraîne une pénalité de  $P > 0$  cases. Quelles modifications faut-il apporter à celle proposée en 3.1) ? Comment peut-on trouver le chemin le moins pénalisant dans cette version du problème du labyrinthe ?

**Question 4 : Algorithme de Dijkstra****(15 points)**

4.1) (10 points) Soit le graphe suivant :



Calculez, à l'aide de l'algorithme de Dijkstra (n'utilisant pas de monceau), la distance minimale entre le point A et tous les autres points dans le graphe. Donnez les étapes de votre exécution de l'algorithme dans la table qui suit.

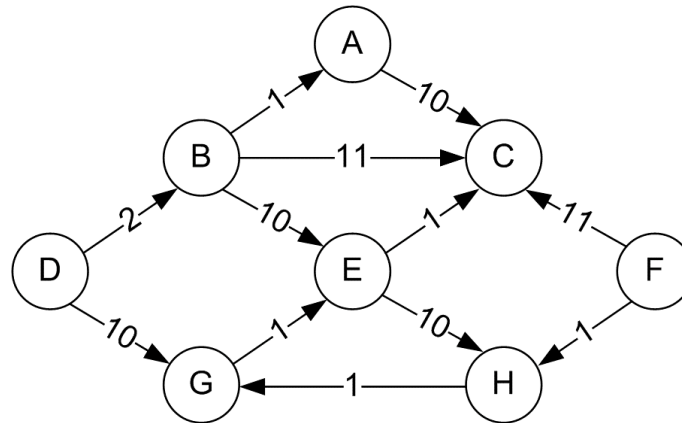
Nœud	Distance A	Distance B	Distance C	Distance D	Distance E	Distance F
-	0	inf	inf	inf	inf	inf
A						

4.2) (5 points) Quelle est la complexité en pire cas de l'algorithme de Dijkstra que vous venez d'exécuter ? Sachant que l'algorithme peut être amélioré, dites comment et donnez la nouvelle complexité ?

**Question 5 : Ordre topologique****(10 points)**

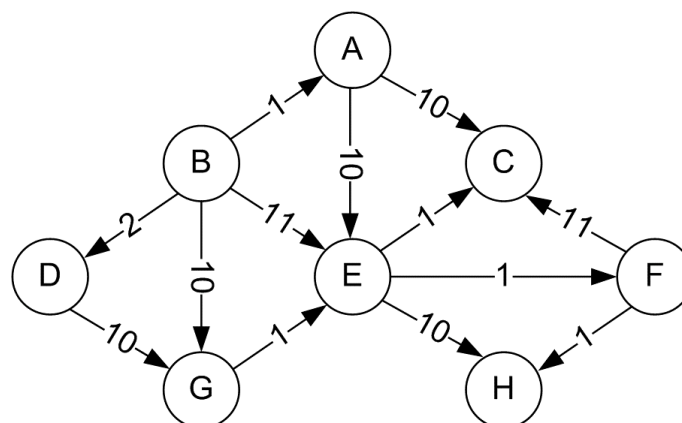
Pour chacun des graphes dirigés aux arêtes pondérées suivants, dites si le graphe admet un ordre topologique et, si oui, proposez-en un. Vous pouvez vous aider des grilles données à l'annexe I. Identifiez clairement cette annexe à votre nom et matricule et assurez-vous de la remettre avec votre copie (même si elle demeure vierge).

5.1) (5 points)



A	B	C	D	E	F	G	H

5.2) (5 points)



A	B	C	D	E	F	G	H

**Question 6 : Rabin-Karp****(15 points)**

L'algorithme Rabin Karp est un algorithme permettant de retrouver une chaîne de caractères dans un texte. Supposons que les chaînes de caractères soient constituées des chiffres 0 à 9 auxquels on associe la valeur numérale de ces derniers (le chiffre 9 vaut 9).

6.1) (2 points) Quel est le nombre de faux-positifs que produit l'exécution de l'algorithme Rabin-Karp sur la chaîne de caractères 96325874124896323495 si l'on recherche le patron 96 en travaillant modulo  $q=11$  ?

**Note :** Vous n'êtes pas obligé de donner vos calculs intermédiaires. On vous conseille cependant de les garder pour les questions suivantes.

6.2) (3 points) Donnez la valeur des sous-séquences correspondant aux faux positifs que vous avez trouvés dans 6.1).

6.3) (2 points) Rabin-Karp produit parfois des faux-positifs (candidats retenus ne remplissant pas le critère de correspondance). Expliquez pourquoi il ne génère pas de faux-négatifs (candidats rejetés remplissant le critère de correspondance).



6.4) On veut rechercher deux patrons  $P_1$  et  $P_2$  de longueur  $m$  dans un texte  $T$  de longueur  $n$ , et ce, en exécutant une variante de l'algorithme Rabin-Karp (appliqué à deux patrons de même longueur et non plus un).

6.4.1) (2 points) Quelle est la complexité (en pire cas) de cet algorithme ? Justifiez brièvement.

6.4.2) (2 points) Combien de faux-positifs produit cet algorithme sur la chaîne de caractères de la question 6.1 (96325874124896323495) en recherchant les patrons 23 et 87 ?

6.4.3) (2 points) Combien de faux-négatifs produit cet algorithme en recherchant les mêmes patrons ?

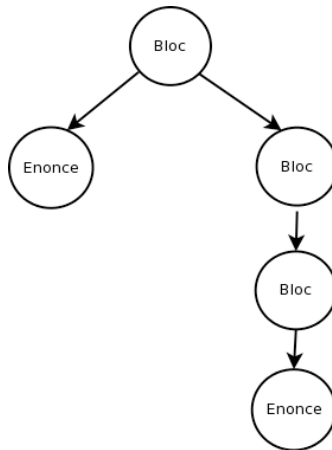
6.4.4) (2 points) Quels sont les décalages retournés pour chacun de ces patrons ?

**Question 7 : Métriques dans les arbres de syntaxe abstraits (10 points)**

Un AST est une représentation abstraite de la syntaxe du code d'une application. Pour simplifier le problème, on ne représente ici que les blocs et les énoncés dans le code. Les blocs peuvent contenir d'autres blocs ou des énoncés, mais les énoncés ne peuvent avoir aucun enfant. Par exemple, le code :

```
public int uneMethode() {  
    int i = 0;  
    while(i < 12) {  
        if(i > 4) {  
            System.out.println(i);  
        }  
    }  
}
```

serait représenté par l'AST suivant :



On désire connaître le nombre d'énoncés et de blocs contenus à l'intérieur de chacun des blocs de l'AST. Par exemple, le bloc racine (qui correspond au bloc `uneMethode`) contient 2 blocs et 2 énoncés. Proposez un algorithme en temps linéaire qui permet de calculer les métriques de chacun des blocs de l'AST. ***Indice*** : Il faut parcourir tous les nœuds du graphe pour faire ce calcul.

**Question 8 : Programmation dynamique****(10 points)**

Les coefficients binomiaux ont une grande utilité en mathématique puisqu'on les retrouve par exemple dans le calcul des combinatoires ou le binôme de Newton. La combinatoire  $k$  de  $n$  ( $k \leq n$ ) est donnée par :

$$\binom{n}{k} = C_n^k = \frac{n!}{k!(n-k)!}$$

Il est également possible de l'exprimer par une formule de récurrence :

$$\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$$

où  $C_0^0 = C_n^0 = C_n^n = 1$ . Cela peut se traduire par l'implémentation suivante :

```
public static int Combinatoire(int n, int k)
{
    if (n < 0 || k < 0)
        return -1;

    if (n == k || k == 0)
        return 1;
    else
        return Combinatoire(n-1, k) + Combinatoire(n-1, k-1);
}
```

8.1) (3 points) En vous appuyant sur la fonction donnée ci-dessus, illustrez le fait que l'expression récursive des coefficients binomiaux présente un cas de recouvrement des sous-problèmes en vous basant sur les étapes de calcul de  $C_4^2$ .

8.2) On vous propose l'implémentation d'un algorithme de programmation dynamique suivante pour le calcul des coefficients binomiaux :

```
public static int CombinatoireDynamique(int n, int k)
{
    if(n < 0 || k < 0)
        return -1;

    int[][] M = new int[ n+1 ][ k+1 ];

    for(int i=0; i<=n; i++)
    {
        for(int j=0; j <= (i<k ? i : k) ; j++)
        {
            if(i==j || j==0)
                M[ i ][ j ] = 1;
            else
                M[ i ][ j ] = M[ i-1 ][ j ] + M[ i-1 ][ j-1 ];
        }
    }

    return M[ n ][ k ];
}
```

8.2.1) (3 points) Détaillez les étapes d'exécution de cet algorithme pour le calcul de  $C_4^2$


8.2.2) (2 points) Peut-on réduire l'espace mémoire nécessaire pour le tableau  $M[n+1][k+1]$  ? Justifiez votre réponse.

8.2.3) (2 points) Quelle est selon vous la plus petite taille (en fonction des paramètres  $n$  et  $k$ ) de la table  $M[][]$  qui permette l'implémentation d'un algorithme de programmation dynamique pour le calcul des coefficients binomiaux ? Justifiez.

## Annexe I

Nom et prénom : \_\_\_\_\_

Matricule : \_\_\_\_\_

	Arcs incidents							
A								
B								
C								
D								
E								
F								
G								
H								
Entrée								
Sortie								

	Arcs incidents							
A								
B								
C								
D								
E								
F								
G								
H								
Entrée								
Sortie								