

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	20103
<i>Professeur</i>		<i>Local</i>	<i>Téléphone</i>
Ettore Merlo – responsable / Tarek Ould Bachir – chargé de cours		M-4105	5758 / 5193
<i>Jour</i>	<i>Date</i>	<i>Durée</i>	<i>Heures</i>
Jeudi	9 décembre 2010	2h30	13h30 - 16h00

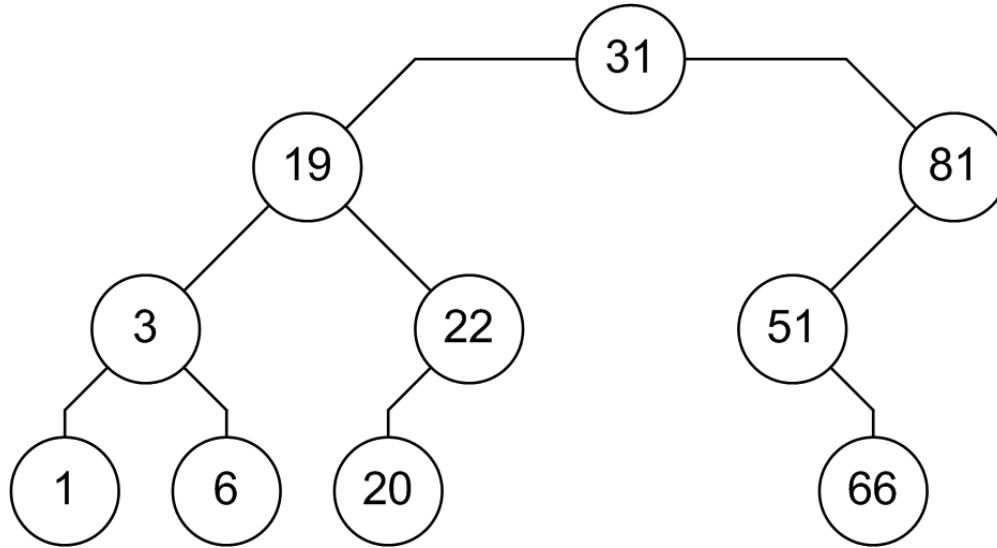
<i>Documentation</i>	<i>Calculatrice</i>	
<input checked="" type="checkbox"/> Aucune <input type="checkbox"/> Toute <input type="checkbox"/> Voir directives particulières	<input type="checkbox"/> Aucune <input type="checkbox"/> Toutes <input checked="" type="checkbox"/> Non programmable	Les cellulaires,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

<i>Directives particulières</i>
<div style="text-align: right;"><i>Bon succès à tous!</i></div>

Important	Cet examen contient <input type="text" value="7"/> questions sur un total de <input type="text" value="18"/> pages (excluant cette page)
	La pondération de cet examen est de <input type="text" value="40"/> %
	Vous devez répondre sur : <input type="checkbox"/> le questionnaire <input type="checkbox"/> le cahier <input checked="" type="checkbox"/> les deux
	Vous devez remettre le questionnaire : <input checked="" type="checkbox"/> oui <input type="checkbox"/> non

Question 1 : Arbre Splay**(14 points)****Note :** Les rotations de cette question sont identiques à celles utilisées avec les arbres AVL.

a) Considérez l'arbre Splay suivant et répondez aux énoncés par vrai ou par faux.



a.1) (2 pts) Ceci est un arbre qui a la structure d'un arbre complet :

Vrai	<input type="checkbox"/>
Faux	<input type="checkbox"/>

a.2) (2 pts) Ceci est un arbre qui a la structure d'un arbre de recherche :

Vrai	<input type="checkbox"/>
Faux	<input type="checkbox"/>

a.3) (2 pts) Ceci est un arbre qui a la structure d'un arbre AVL :

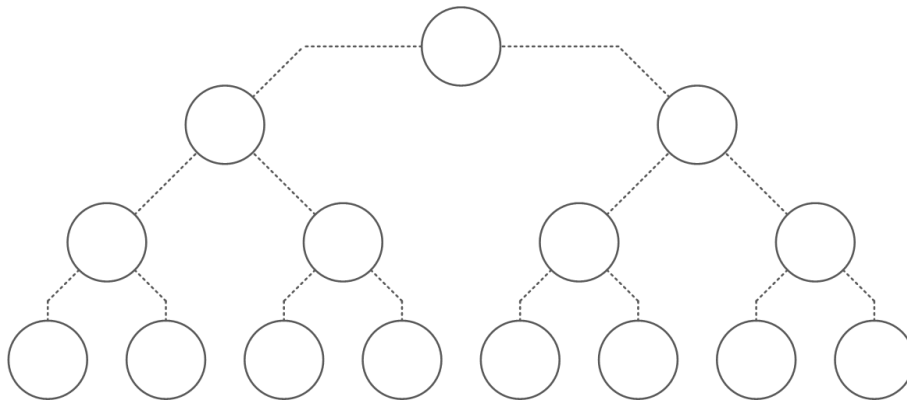
Vrai	<input type="checkbox"/>
Faux	<input type="checkbox"/>

b) Proposez une modification à l'arbre précédent qui le transforme pour rencontrer :

b.1.1) (1 pt) Pour que cet arbre ait la structure d'un arbre complet, il suffit :

- | | |
|--------------------------|--|
| <input type="checkbox"/> | de rien faire, cet arbre a déjà la structure d'un arbre complet |
| <input type="checkbox"/> | d'effectuer une rotation simple à droite autour du nœud 51 |
| <input type="checkbox"/> | d'effectuer une rotation double, à droite autour du nœud 51, à gauche autour de 81 |
| <input type="checkbox"/> | d'effectuer une rotation simple à gauche autour du nœud 51 |
| <input type="checkbox"/> | d'effectuer une rotation double à gauche autour du nœud 51, à droite autour de 81 |

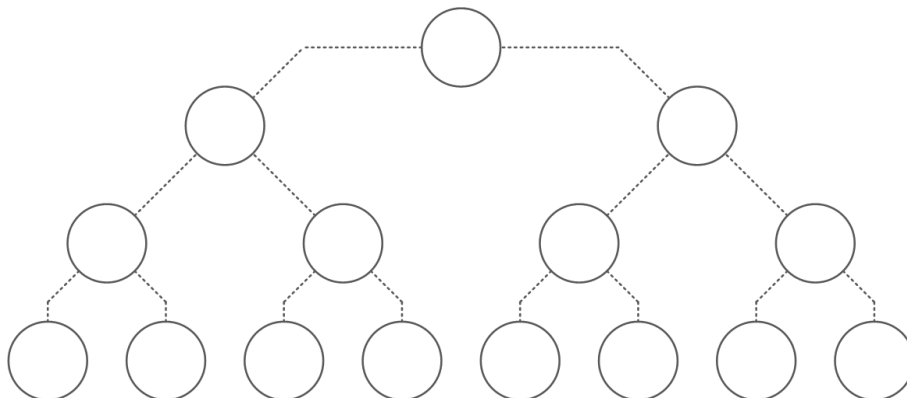
b.1.2) (1 pt) Dessinez l'arbre ainsi obtenu :



b.2.1) (1 pt) Pour que cet arbre ait la structure d'un arbre de recherche, il suffit :

- | | |
|--------------------------|--|
| <input type="checkbox"/> | de rien faire, cet arbre a déjà la structure d'un arbre de recherche |
| <input type="checkbox"/> | d'effectuer une rotation simple à droite autour du nœud 51 |
| <input type="checkbox"/> | d'effectuer une rotation double, à droite autour du nœud 51, à gauche autour de 81 |
| <input type="checkbox"/> | d'effectuer une rotation simple à gauche autour du nœud 51 |
| <input type="checkbox"/> | d'effectuer une rotation double à gauche autour du nœud 51, à droite autour de 81 |

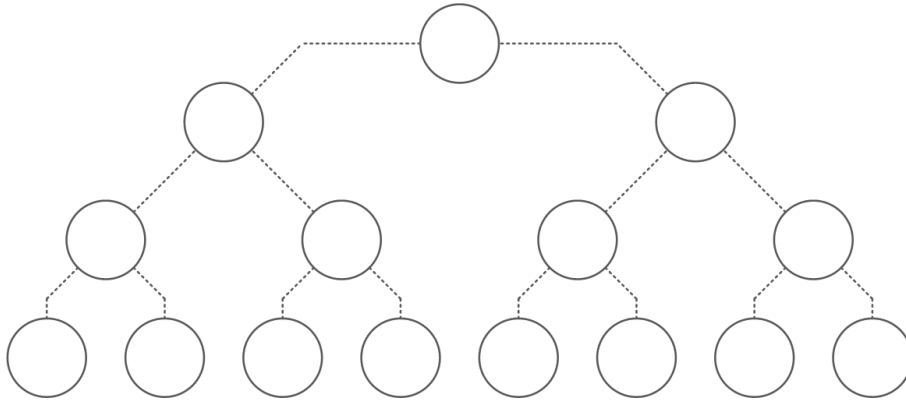
b.2.2) (1 pt) Dessinez l'arbre ainsi obtenu :



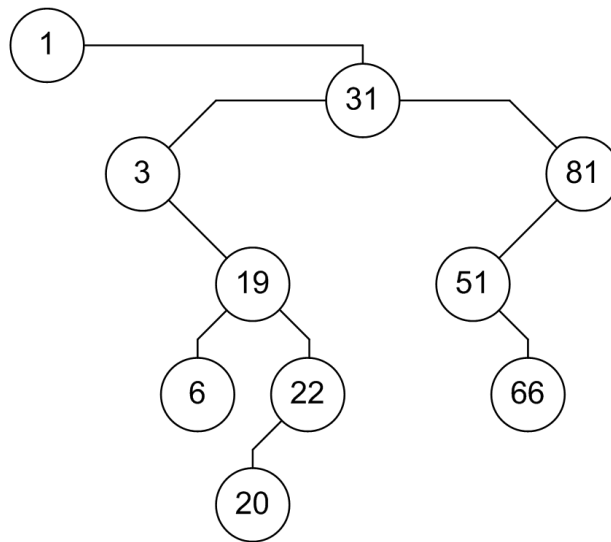
b.3.1) (1 pt) Pour que cet arbre ait la structure d'un arbre AVL, il suffit :

- | | |
|--------------------------|--|
| <input type="checkbox"/> | de rien faire, cet arbre a déjà la structure d'un arbre AVL |
| <input type="checkbox"/> | d'effectuer une rotation simple à droite autour du nœud 51 |
| <input type="checkbox"/> | d'effectuer une rotation double, à droite autour du nœud 51, à gauche autour de 81 |
| <input type="checkbox"/> | d'effectuer une rotation simple à gauche autour du nœud 51 |
| <input type="checkbox"/> | d'effectuer une rotation double à gauche autour du nœud 51, à droite autour de 81 |

b.3.2) (1 pt) Dessinez l'arbre ainsi obtenu :



c) On effectue un get(1) sur l'arbre Splay du départ et on obtient ce qui suit.

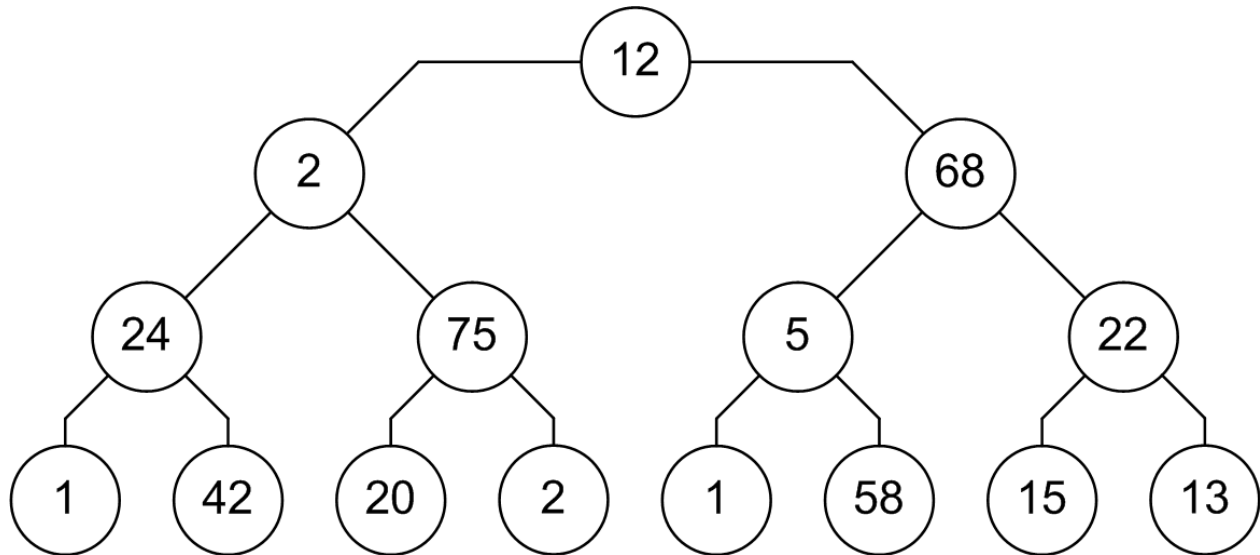


c.1) (1 pt) Expliquez l'intérêt des arbres Splay s'ils produisent des arbres aussi mal débalancés.

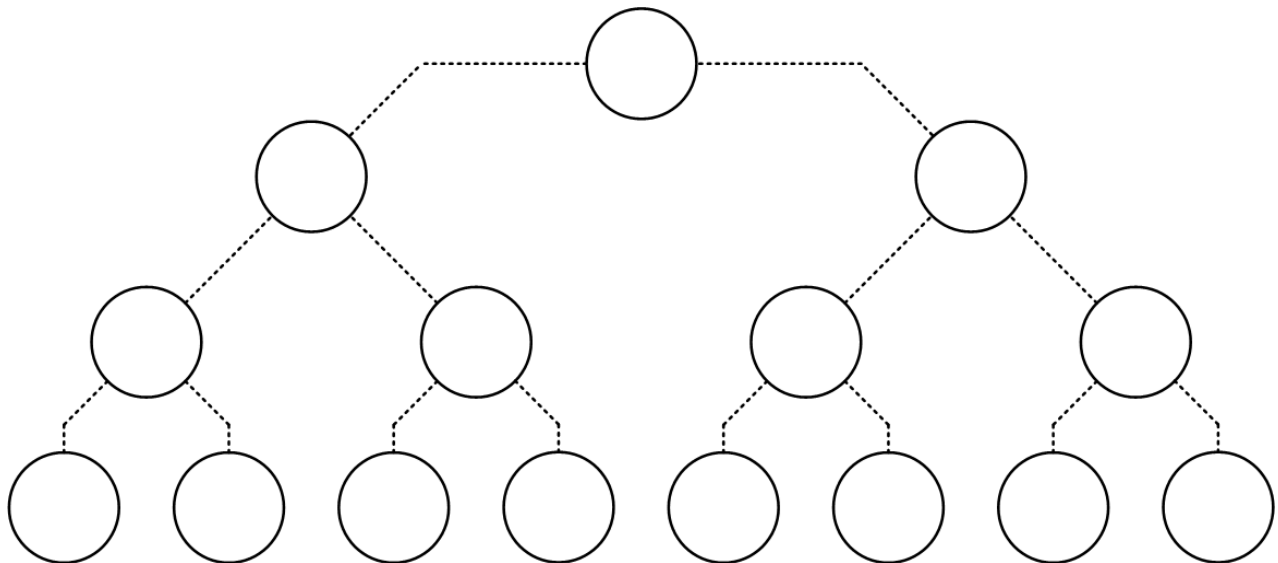
c.2) (1 pt) Pensez-vous que ce résultat (l'arbre obtenu) soit correct ? Discutez.

Question 2 : Monceaux**(16 points)**

- a) (4 pts) Construisez, selon la technique vue dans le cours, un monceau à partir de l'arbre binaire suivant :

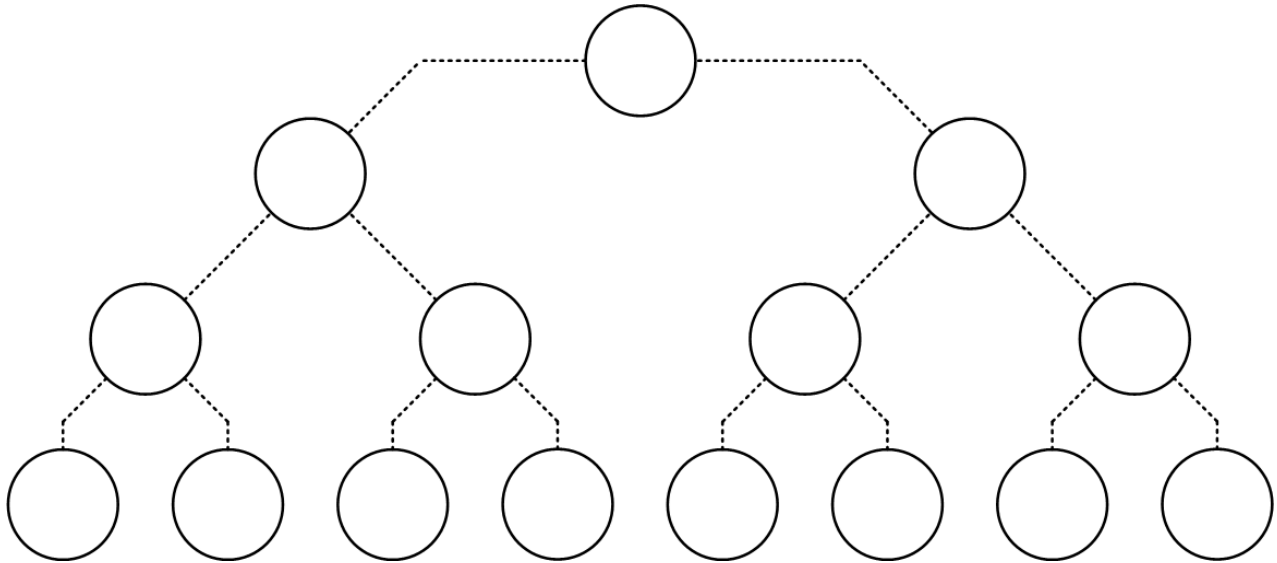


Monceau résultant :

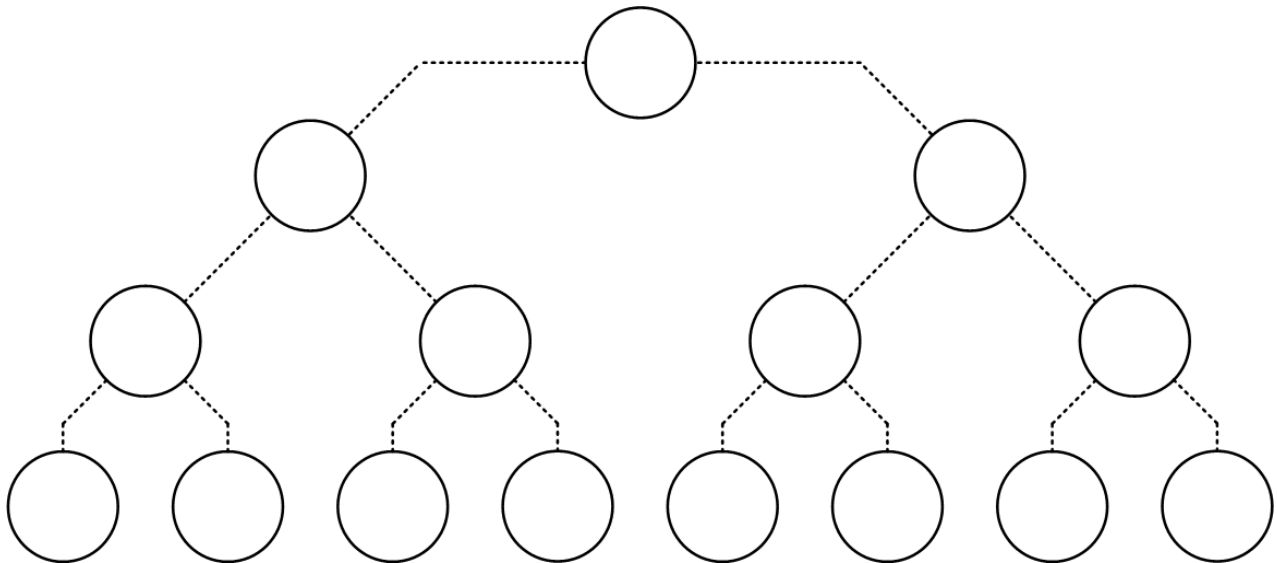


b) Dessinez l'état de ce monceau après deux appels consécutifs à `deleteMin()` :

b.1) (3 pts) Monceau résultant du premier `deleteMin()`

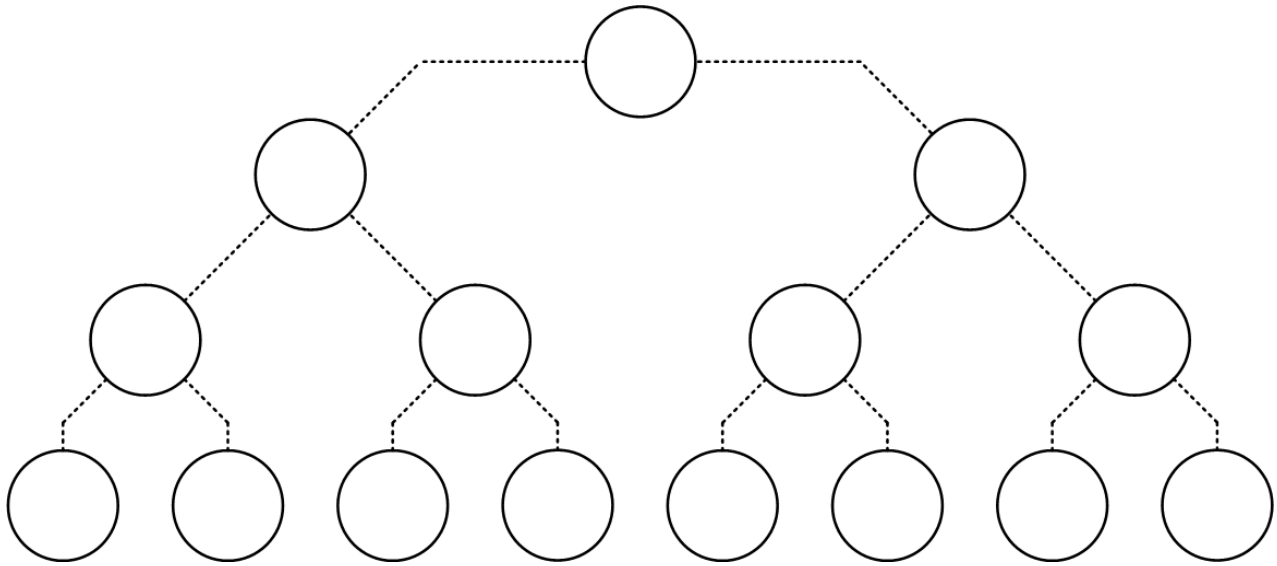


b.2) (3 pts) Monceau résultant du second `deleteMin()`



c) Dessinez l'état du dernier monceau obtenu auquel on insère la clé 19

c.1) (3 pts) Monceau résultant de insert(19)



c.2) (3 pts) Dessinez l'état du tableau contenant le monceau résultant de l'insertion de 19 :

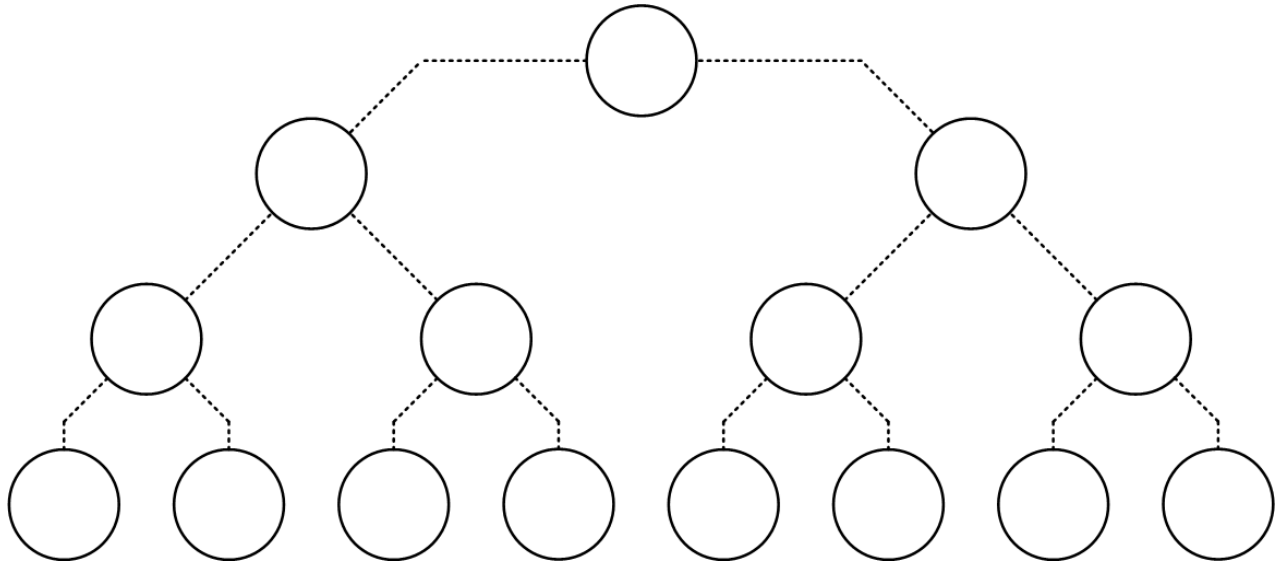
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Question 3 : Tri par monceau**(10 points)**

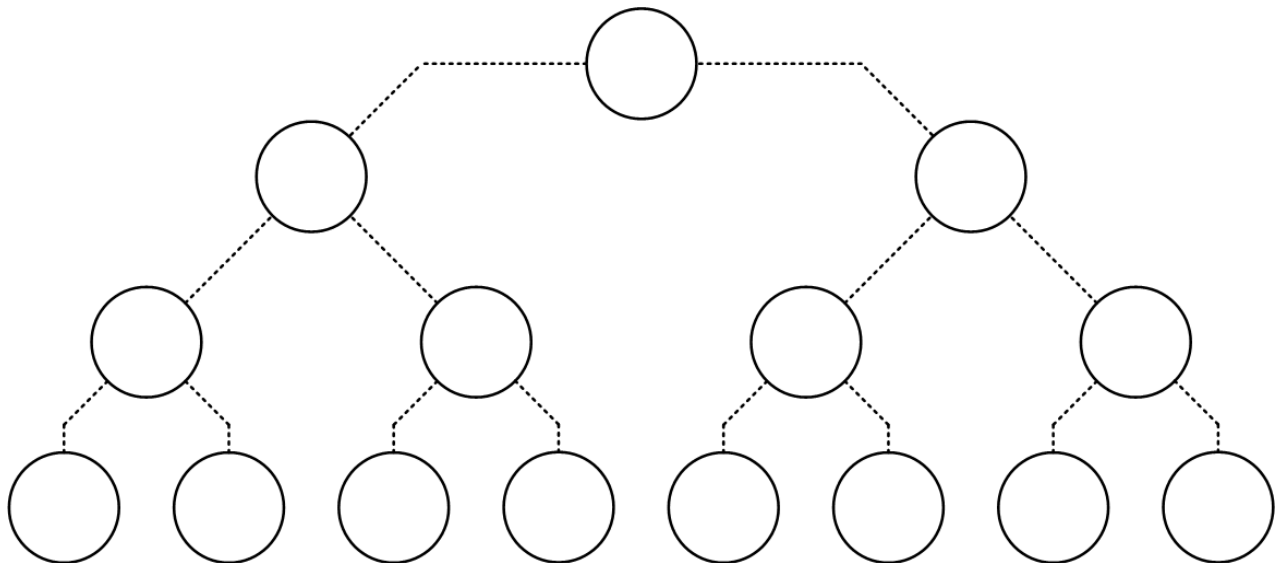
On désire trouver les trois plus grands éléments d'un vecteur d'entiers. On vous suggère, pour ce faire, d'utiliser un monceau MAX et de trouver les trois plus grands éléments au niveau de la racine et de ses deux enfants.

a) Illustrez la pertinence de cette stratégie en construisant un monceau MAX à partir de chacun des vecteurs suivants :

a.1) (3 pts) 12, 2, 68, 24, 75, 5, 22, 1, 58, 20, 2, 1, 42, 15, 13



a.2) (3 pts) 12, 2, 68, 24, 75, 5, 22, 1, 42, 20, 2, 1, 58, 15, 13



- b) **(2 pts)** Commentez cette stratégie. Est-elle justifiée ou non ? Expliquez clairement votre réponse.
- c) **(2 pts)** Proposez une stratégie que vous jugez plus efficace. Justifiez votre réponse.

Question 4 : DP-Matching**(10 points)**

- a) (4 pts) En utilisant le tableau 4.1, retrouvez la plus longue sous-séquence commune aux chaînes d'entrée $X = \text{« ORANAISES »}$ et $Y = \text{« CHAGRINEUSES »}$:

Tableau 4.1

	Y	C	H	A	G	R	I	N	E	U	S	E	S
X	0	0	0	0	0	0	0	0	0	0	0	0	0
O	0												
R	0												
A	0												
N	0												
A	0												
I	0												
S	0												
E	0												
S	0												

Plus longue sous-séquence commune et sa longueur:

- b) (6 pts) Proposez trois autres *PLSC* des entrées $X = \text{« ORANAISES »}$ et $Y = \text{« CHAGRINEUSES »}$:

Question 5 : PMA automate de reconnaissance de motifs
(pattern matching)

(15 points)

Considérez du texte de longueur variable défini sur l'alphabet $SIGMA = \{n,a,p\}$ de cardinalité égale à trois, où le symbole a représente les caractères alphabétiques, n représente un chiffre numérique et p représente le caractère '.'.

- a) **(5 pts)** Construisez l'automate de reconnaissance des motifs capable de reconnaître une séquence de trois symboles n (nnn) dans un texte défini sur l'alphabet SIGMA mentionné. Remplissez le tableau 5.1.

Tableau 5.1

q	c	$\delta(a,c)$	État initial (V/F)	État final (V/F)
0	n			
0	a			
0	p			
1	n			
1	a			
1	p			
2	n			
2	a			
2	p			
3	n			
3	a			
3	p			
4	n			
4	a			
4	p			
5	n			
5	a			

5	p			
6	n			
6	a			
6	p			
7	n			
7	a			
7	p			
8	n			
8	a			
8	p			
9	n			
9	a			
9	p			
10	n			
10	a			
10	p			
11	n			
11	a			
11	p			
12	n			
12	a			
12	p			

- b) (10 pts) Construisez l'automate de reconnaissance des motifs capable de reconnaître une séquence de trois symboles n suivis par un symbole p suivi par trois nouveaux symboles n (nnnpnnn) dans un texte défini sur l'alphabet SIGMA mentionné. Remplissez le tableau 5.2.

Tableau 5.2

q	c	$\delta(a,c)$	État initial (V/F)	État final (V/F)
0	n			
0	a			
0	p			
1	n			
1	a			
1	p			
2	n			
2	a			
2	p			
3	n			
3	a			
3	p			
4	n			
4	a			
4	p			
5	n			
5	a			
5	p			
6	n			
6	a			
6	p			

7	n			
7	a			
7	p			
8	n			
8	a			
8	p			
9	n			
9	a			
9	p			
10	n			
10	a			
10	p			
11	n			
11	a			
11	p			
12	n			
12	a			
12	p			

Question 6 : RB : Rabin-Karp**(10 points)**

Considérez les chaînes de caractères correspondantes au texte de longueur variable défini sur l'alphabet $\text{SIGMA} = \{a,b\}$ de cardinalité égale à deux.

Considérez le codage Rabin-Karp selon les équations suivantes :

$$1) \text{RK}_{s+1} = (d(\text{RK}_s - h \cdot T[s+1]) + T[m+s+1]) \bmod q$$

$$2) h = d^{m-1} \bmod q$$

$$3) d = 256$$

$$4) q = 7$$

Simplifiez les équations 1, 2, 3 et 4 en les modifiant pour accélérer et rendre encore plus efficace le calcul de la position de concordance pour les motifs suivants :

a) (5 pts) Transition entre la fin d'une séquence de a et le début d'une séquence de b.
 (...ab..., ...aab, ...abb..., ...aabb..., etc.)
 Écrivez les équations modifiées a1, a2, a3, a4.

b) (5 pts) Transition entre la fin d'une séquence de b et le début d'une séquence de a.
 (...ba..., ...bba..., ...baa..., ...bbba..., etc.)
 Écrivez les équations modifiées b1, b2, b3, b4.

Question 7 : RB : Graphes**(25 points)**a) **(10 pts)** Considérez le graphe $G_1 = (V_1, E_1)$ suivant

$$V_1 = \{0, 1, 2, 3, 4, 5, 6, 7\}$$

$$E_1 = \{(0,1), (1,2), (1,6), (2,0), (2,6), (2,5), \\ (3,0), (4,5), (5,7), (6,5), (6,7)\}$$

Remplissez le tableau 7.1 en calculant l'arbre sous-tendant minimum selon l'algorithme de Prim (*minimum spanning tree*) en partant du nœud $v=0$.

Tableau 7.1

v	d_v	p_v
0		
1		
2		
3		
4		
5		
6		
7		

b) **(15 pts)** Considérez le graphe $G_2 = (V_2, E_2)$ suivant

$$V_2 = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$$

$$E_2 = \{(0,3), (1,2), (1,0), (2,0), (3,2), (4,5), (4,1), (5,1), (5,2), (5,6), (5,7), (6,7), (6,3), (6,2), (7,4), (8,4), (8,7)\}$$

En utilisant l'algorithme de calcul des composantes fortement connexes (CFC) reporté en annexe à la page 20, remplissez les tableaux 7.2, 7.3 et 7.4. Remplissez le tableau 7.5 en faisant partir de zéro les identificateurs des CFC.

Tableau 7.2

v	f[u]
0	
1	
2	
3	
4	
5	
6	
7	
8	

Tableau 7.3 - Ensemble des nœuds V_T

w	0	1	2	3	4	5	6	7	8	9	10	11	12
$w \in V_T$ (V/F)													

Tableau 7.4**E_T**

Nœud départ \ Nœud arrivé														
	0	1	2	3	4	5	6	7	8	9	10	11	12	
0	○	○	○	○	○	○	○	○	○	○	○	○	○	
1	○	○	○	○	○	○	○	○	○	○	○	○	○	
2	○	○	○	○	○	○	○	○	○	○	○	○	○	
3	○	○	○	○	○	○	○	○	○	○	○	○	○	
4	○	○	○	○	○	○	○	○	○	○	○	○	○	
5	○	○	○	○	○	○	○	○	○	○	○	○	○	
6	○	○	○	○	○	○	○	○	○	○	○	○	○	
7	○	○	○	○	○	○	○	○	○	○	○	○	○	
8	○	○	○	○	○	○	○	○	○	○	○	○	○	
9	○	○	○	○	○	○	○	○	○	○	○	○	○	
10	○	○	○	○	○	○	○	○	○	○	○	○	○	
11	○	○	○	○	○	○	○	○	○	○	○	○	○	
12	○	○	○	○	○	○	○	○	○	○	○	○	○	

Tableau 7.5 - Composantes fortement connexes

v	0	1	2	3	4	5	6	7	8	9	10	11	12
CFC													

Annexe

PP(G)

```

1  pour chaque sommet  $u \in S[G]$ 
2      faire  $\text{couleur}[u] \leftarrow \text{BLANC}$ 
3           $\pi[u] \leftarrow \text{NIL}$ 
4   $\text{date} \leftarrow 0$ 
5  pour chaque sommet  $u \in S[G]$ 
6      faire si  $\text{couleur}[u] = \text{BLANC}$ 
7          alors VISITER-PP( $u$ )

```

VISITER-PP(u)

```

1   $\text{couleur}[u] \leftarrow \text{GRIS} \triangleright$  sommet blanc  $u$  vient d'être découvert.
2   $\text{date} \leftarrow \text{date} + 1$ 
3   $d[u] \leftarrow \text{date}$ 
4  pour chaque  $v \in \text{Adj}[u] \triangleright$  Exploration de l'arc  $(u, v)$ .
5      faire si  $\text{couleur}[v] = \text{BLANC}$ 
6          alors  $\pi[v] \leftarrow u$ 
7              VISITER-PP( $v$ )
8   $\text{couleur}[u] \leftarrow \text{NOIR} \triangleright$  noircir  $u$ , car on en a fini avec lui.
9   $f[u] \leftarrow \text{date} \leftarrow \text{date} + 1$ 

```

COMPOSANTES-FORTEMENT-CONNEXES(G)

- 1 appeler PP(G) pour calculer les dates de fin de traitement $f[u]$
pour chaque sommet u
- 2 calculer ${}^T G$
- 3 appeler PP(${}^T G$), mais dans la boucle principale de PP, traiter les sommets
par ordre de $f[u]$ (calculés en ligne 1) décroissants
- 4 imprimer les sommets de chaque arborescence de la forêt obtenue
en ligne 3 en tant que composante fortement connexe distincte