



POLYTECHNIQUE
MONTRÉAL

Corrigé examen final

INF2010

Sigle du cours

Identification de l'étudiant(e)		
Nom :	Prénom :	
Signature :	Matricule :	Groupe :

Sigle et titre du cours		Groupe	Trimestre
INF2010 – Structures de données et algorithmes		Tous	20141
Professeur		Local	Téléphone
Ettore Merlo, responsable / Tarek Ould Bachir, chargé		M-5028	7128 / 5193
Jour	Date	Durée	Heures
Samedi	19 avril 2014	2h30	13h30-16h00
Documentation		Calculatrice	
<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.
Directives particulières			
Un cahier supplémentaire vous sera remis. Servez-vous de ce cahier comme brouillon. Toutes vos réponses doivent être faites sur le questionnaire. Le cahier supplémentaire n'est pas à remettre à la fin de l'examen.			
Important	Cet examen contient <input type="text" value="6"/> questions sur un total de <input type="text" value="15"/> pages (excluant cette page)		
	La pondération de cet examen est de <input type="text" value="40"/> %		
	Vous devez répondre sur : <input checked="" type="checkbox"/> le questionnaire <input type="checkbox"/> le cahier <input type="checkbox"/> les deux		
	Vous devez remettre le questionnaire : <input checked="" type="checkbox"/> oui <input type="checkbox"/> non		

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

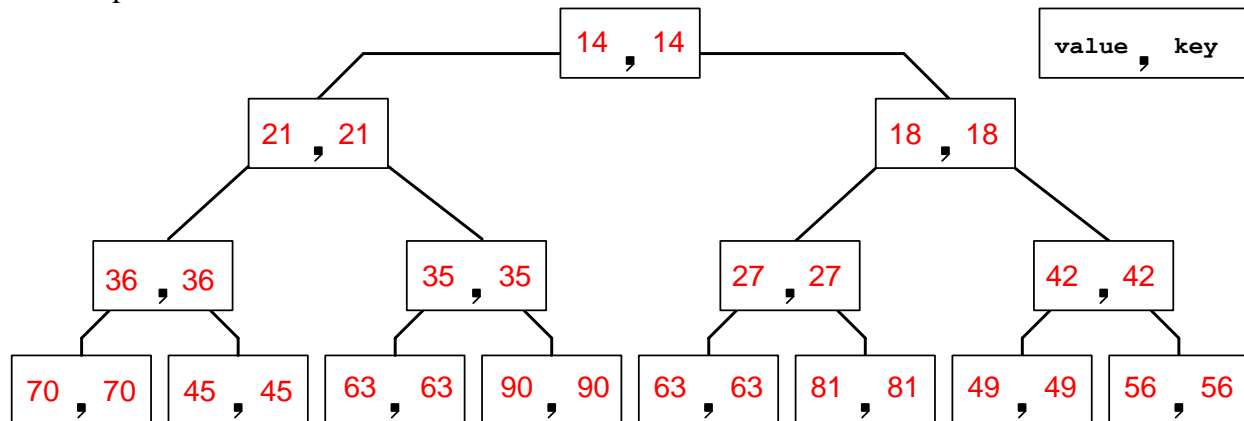
Question 1 : Monceaux**(20 points)**

Pour cette question, vous pouvez vous référer au code Java de l'Annexe 1.

- a) (2 pts) Dessinez le monceau contenu en mémoire dans le tableau ci-après. Indiquez dans les cases les valeurs de **value** et **key** de l'objet **Entry** contenues dans le monceau.

Indice	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Contenu	-	14	21	18	36	35	27	42	70	45	63	90	63	81	49	56

Votre réponse :



- b) Considérez la fonction `changeKey(int location, int newKey)` donnée à l'Annexe 1 permettant de modifier la clé d'une entrée du monceau.

b.1) (2 pts) Quelle est la complexité asymptotique de cette fonction en pire cas ? Justifiez clairement votre réponse. Une réponse non justifiée ne sera pas considérée.

En pire cas, cette fonction a une complexité $O(\lg(n))$ car la percolation vers le bas et l'insertion ont un pire cas en $O(\lg(n))$.

b.2) (2 pts) Quelle est la complexité asymptotique de cette fonction en meilleur cas ? Justifiez clairement votre réponse. Une réponse non justifiée ne sera pas considérée.

En meilleur cas, cette fonction a une complexité $O(1)$ car la percolation vers le bas et l'insertion ont un meilleur cas en $O(1)$. Ce cas survient par exemple lorsqu'on change la clé de la dernière entrée pour une clé supérieure ou égale à sa clé courante.

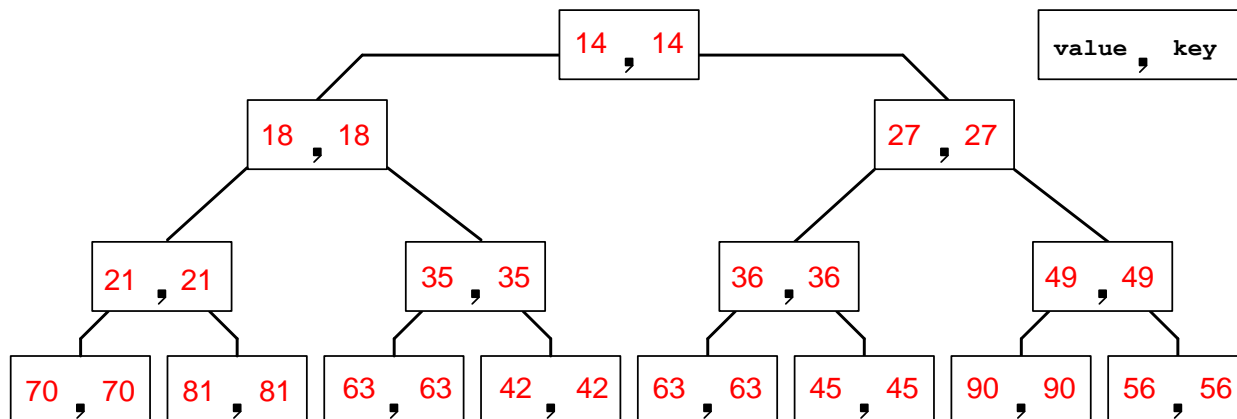
- c) (3 pts) En vous fiant au code donné à l'Annexe 1, dessinez le monceau résultant de l'appel : `BinaryHeap(values_1, true)`

Indiquez dans les cases les valeurs de **value** et **key** de l'objet **Entry** contenues dans le monceau.

Le tableau **values_1** est le suivant:

Indice	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Contenu	36	18	45	81	63	27	90	70	21	35	42	63	14	49	56

Monceau résultant :



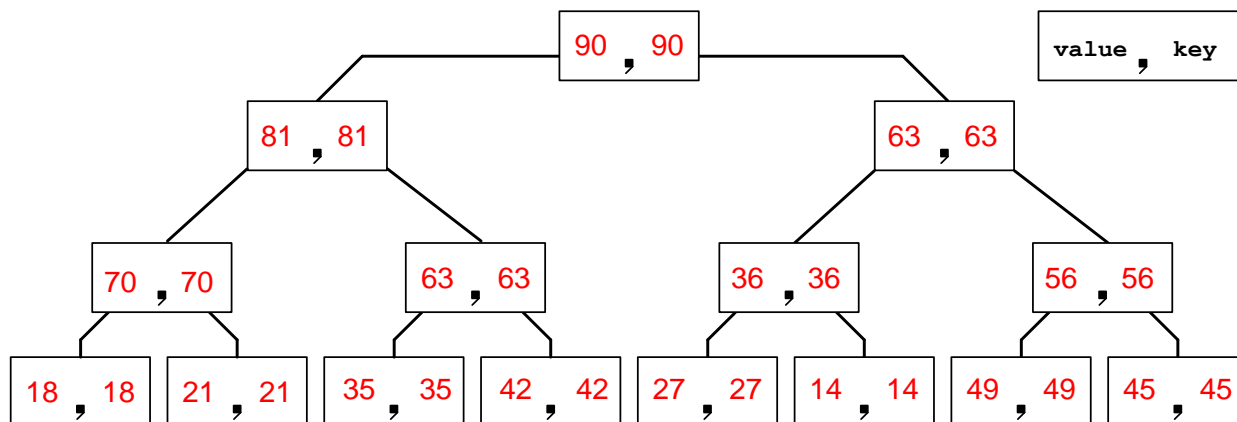
- d) (3 pts) En vous fiant au code donné à l'Annexe 1, dessinez le monceau résultant de l'appel : `BinaryHeap(values_1, false)`

Indiquez dans les cases les valeurs de **value** et **key** de l'objet **Entry** contenues dans le monceau.

Le tableau **values_1** est le suivant:

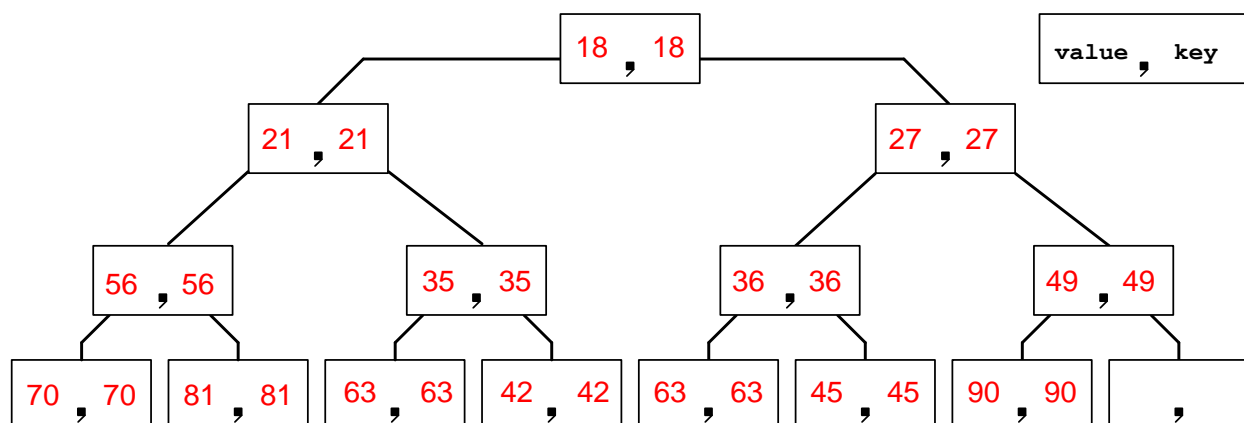
Indice	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Contenu	36	18	45	81	63	27	90	70	21	35	42	63	14	49	56

Monceau résultant :

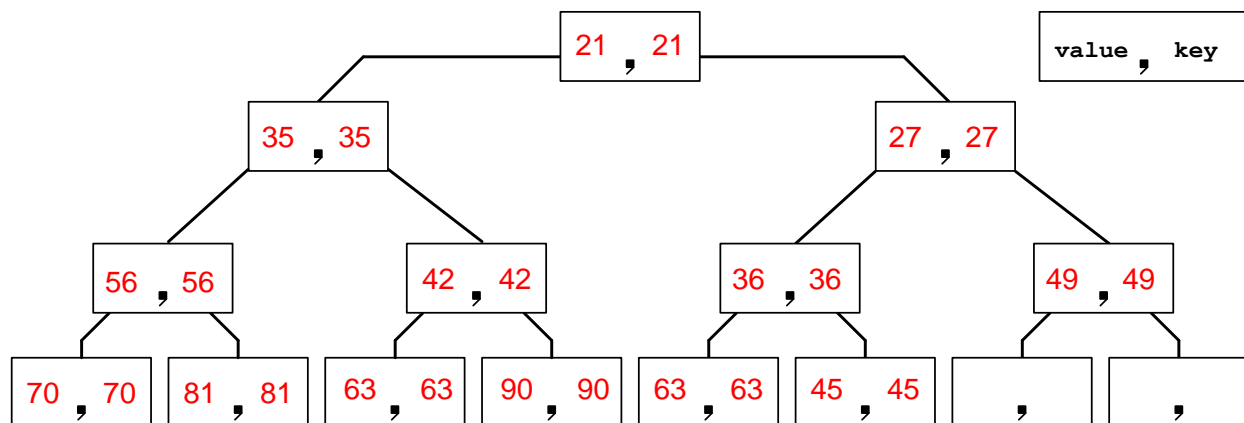


e) Dessinez l'état du monceau de la question 1.c) suite à deux appels consécutifs à `deleteRoot()` :

e.1) (1 pts) Monceau résultant du premier `deleteRoot()`. Indiquez dans les cases les valeurs de **value** et **key** de l'objet **Entry** contenues dans le monceau.

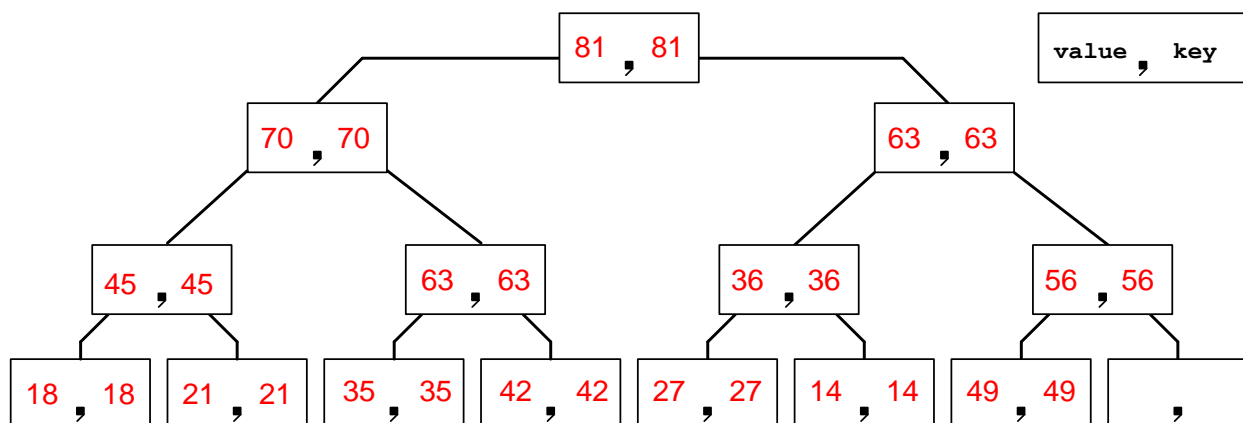


e.2) (1 pts) Monceau résultant du second `deleteRoot()`. Indiquez dans les cases les valeurs de **value** et **key** de l'objet **Entry** contenues dans le monceau.

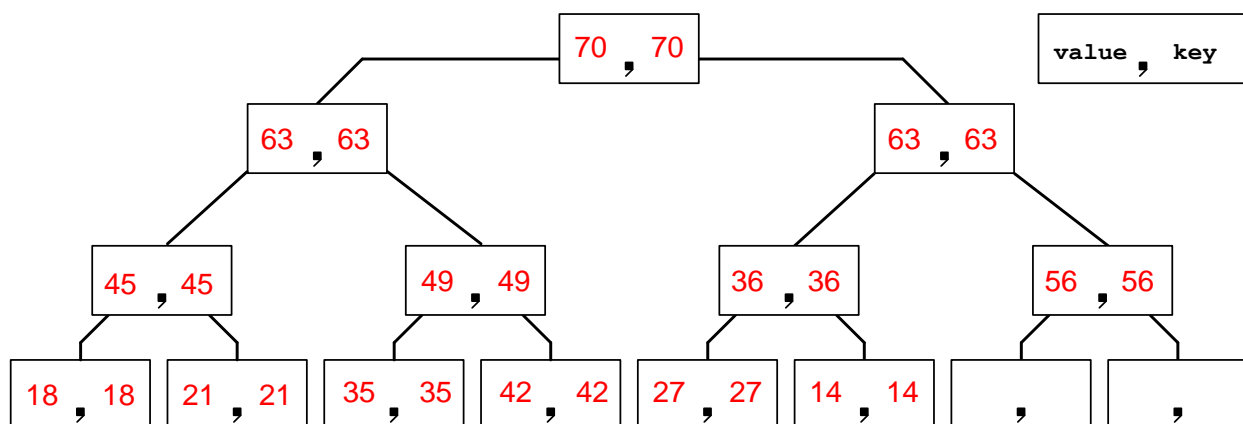


f) Dessinez l'état du monceau de la question 1.d) suite à deux appels consécutifs à `deleteRoot()` :

f.1) (1 pts) Monceau résultant du premier `deleteRoot()`. Indiquez dans les cases les valeurs de **value** et **key** de l'objet **Entry** contenues dans le monceau.



f.2) (1 pts) Monceau résultant du second `deleteRoot()`. Indiquez dans les cases les valeurs de **value** et **key** de l'objet **Entry** contenues dans le monceau.

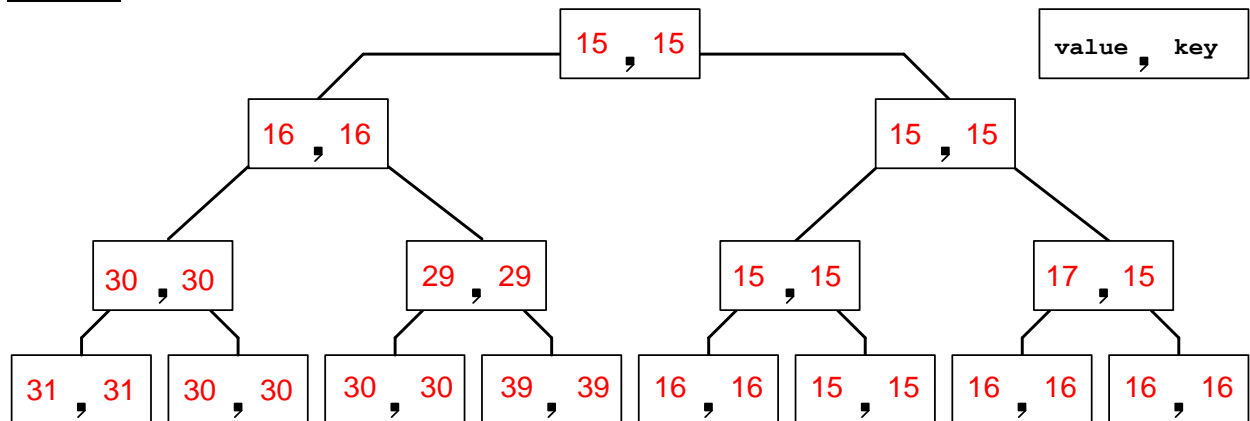


g) (4 pts) Exécutez `changeKey(15, 15)` sur le monceau suivant construit après un appel : `BinaryHeap(values_2)`. Indiquez dans les cases les valeurs de **value** et **key** de l'objet **Entry** contenues dans le monceau.

Le tableau `values_2` est le suivant:

Indice	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Contenu	15	16	15	30	29	15	16	31	30	30	39	16	15	16	17

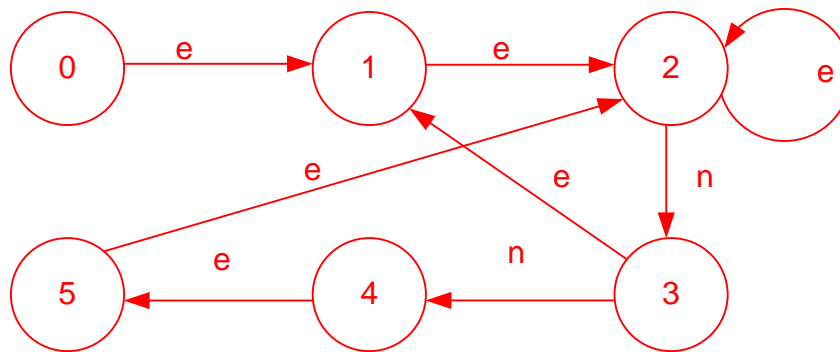
Résultat:



Question 2 : Recherche de patron**(14 points)**

On vous demande de retrouver le patron $P[1:8] = \text{« eenne »}$ dans un texte.

a) (4 pnts) Dessiner le diagramme d'états de l'automate à états finis permettant de ce faire :



Les arcs non représentés renvoient à 0

b) (4 pnts) Donner la table de transitions de l'automate recherché.

q\a	e	n	Autre
0	1	0	0
1	2	0	0
2	2	3	0
3	1	4	0
4	5	0	0
5	2	0	0

- c) (6 pts) Quels seront respectivement le ou les états les plus visités et le ou les états les moins visités par l'automate une fois arrivé à la fin du texte suivante.

$T[1 : 34] = \text{« eeeeeeeennnneneeeennneeeennneee »}$

Aidez-vous de la table suivante:

T[i]	e	e	e	e	e	e	e
q	1	2	2	2	2	2	2
T[i]	e	n	n	n	e	n	e
q	2	3	4	0	1	0	1
T[i]	e	e	e	e	n	n	e
q	2	2	2	2	3	4	5
T[i]	e	e	e	n	n	e	e
q	2	2	2	3	4	5	2
T[i]	e	n	n	e	e	e	
q	2	3	4	5	2	2	

Le ou les états les plus visités : 2, visité 18 fois

Le ou les états les moins visités : 0, visité 2 fois, (Autre réponse acceptée, 0 1 et 5, visités 3 fois si on considère l'état initial de l'automate)

Question 3 : Programmation dynamique**(15 points)**

a) (10 points) En utilisant l'algorithme vu en classe, donnez la longueur de la PLSC des séquences de caractères $X = \text{"abcbcaacbabc"}$ et $Y = \text{"bbacbabcabcabca"}$.

Aidez-vous du tableau donné ci-après. Inscrivez votre réponse à la page suivante.

X\Y		a	b	c	b	c	a	a	c	b	a	b	c
	0	0	0	0	0	0	0	0	0	0	0	0	0
b	0	H, 0	D, 1	G, 1	D, 1					D, 1		D, 1	
b	0		D, 1		D, 2	G, 2				D		D	
a	0	D, 1	H, 1				D, 3	D			D		
c	0			D, 2		D, 3	H, 3		D				D
b	0		D		D, 3	G, 3	H, 3			D		D	
a	0	D					D	D, 4			D		
b	0		D		D			H, 4		D		D	
c	0			D		D			D, 5				D
b	0		D		D					D, 6		D	
c	0			D		D			D	H, 6			D
a	0	D					D	D			D, 7		
b	0		D		D					D		D, 8	
c	0			D		D			D				D, 9
a	0	D					D	D			D		H, 9

Longueur de la PLSC : **bbaacbabc**

PLCS trouvée : **9**

b) (5 pts) Il existe une autre PLSC pour les séquences X et Y que celle trouvée à la question 3.a. Donnez-la:

2^e PLCS : **cbacbabc**

3^e PLCS : **acbabc**

Question 4 : Programmation dynamique**(20 points)**

On désire trouver le parenthésage idéal pour multiplier les matrices A_1 à A_5 permettant de minimiser le nombre de multiplications (scalaires) à effectuer. Les matrices sont dimensionnées comme suit :

$A_1 : 2 \times 1$; $A_2 : 1 \times 1$; $A_3 : 1 \times 3$; $A_4 : 3 \times 1$; $A_5 : 1 \times 6$

Considérez les tables **m** et **s** obtenue par l'exécution de l'algorithme dynamique vu en cours.

m	1	2	3	4	5
1	0	2	8	12	24
2		0	3	10	16
3			0	3	15
4				0	18
5					0

s	1	2	3	4	5
1		1	2	1	4
2			2	2	4
3				3	4
4					4
5					

Compléter cette table pour répondre aux questions suivantes :

Rappel : $m[i, j] = \min\{m[i, k] + m[k+1, j] + p_{i-1}p_kp_j\}$ pour $k = i$ à $j-1$, sachant que la matrice A_i a une dimension $p_{i-1} \times p_i$.

a) (3 pts) Donnez le parenthésage optimal pour multiplier A_1 à A_3 . Donnez son coût.

Parenthésage optimal: (A_1 A_2) A_3

Coût: 8

b) (4 pts) Donnez le parenthésage optimal pour multiplier A_1 à A_4 . Donnez son coût.

Parenthésage optimal: A_1 (A_2 (A_3 A_4))

Coût: 12

c) (5 pts) Donnez le parenthésage optimal pour multiplier A_1 à A_5 . Donnez son coût.

Parenthésage optimal: (A_1 (A_2 (A_3 A_4))) A_5

Coût: 24

d) (8 pts) Si A_1 , A_2 et A_3 étaient des matrices 2×2 , 2×3 et 3×3 respectivement, quel serait le parenthésage optimal pour multiplier A_1 à A_4 ? Quel serait son coût ? Aidez-vous des matrices suivantes pour répondre à la question.

m	1	2	3	4	5
1	0	12	30	19	
2		0	18	15	
3			0	9	
4				0	
5					0

s	1	2	3	4	5
1		1	2	1	
2			2	2	
3				3	
4					4
5					

Parenthésage optimal: $A_1 (A_2 (A_3 A_4))$

Coût: 19

Question 5 : Ordre topologique**(13 points)**

- a) (7 pts) Donnez l'ordre topologique du graphe suivant en appliquant l'algorithme utilisant une file vu en classe.

$$V = \{A, B, C, D, E, F, G\}$$

$$E = \{(A, D), (C, F), (D, F), (D, G), (E, B), (E, C)\}$$

Nœud	1	2	3	4	5	6	7
A	0	-	-	-	-	-	-
B	1	1	0	-	-	-	-
C	1	1	0	-	-	-	-
D	1	0	-	-	-	-	-
E	0	-	-	-	-	-	-
F	2	2	2	1	1	0	-
G	1	1	1	0	-	-	-
Entrée	A, E	D	B, C	G	-	F	-
Sortie	A	E	D	B	C	G	F

Ordre trouvé (débuter la numérotation à 1) :

Nœud	A	B	C	D	E	F	G
Ordre :	1	4	5	3	2	7	6

- b) (6 pts) Ce graphe admet au moins trois autres ordres topologiques. Donnez-les. D'autres tables vous sont fournies à l'Annexe 2. Utilisez-les si nécessaire.

Ordre alternatif #1 :

Nœud	A	B	C	D	E	F	G
Ordre:	1	5	4	3	2	6	7

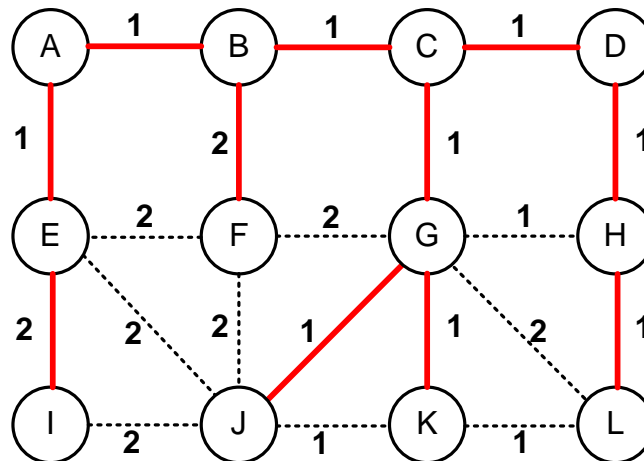
Ordre alternatif #2 :

Nœud	A	B	C	D	E	F	G
Ordre:	2	3	4	5	1	6	7

Ordre alternatif #3 :

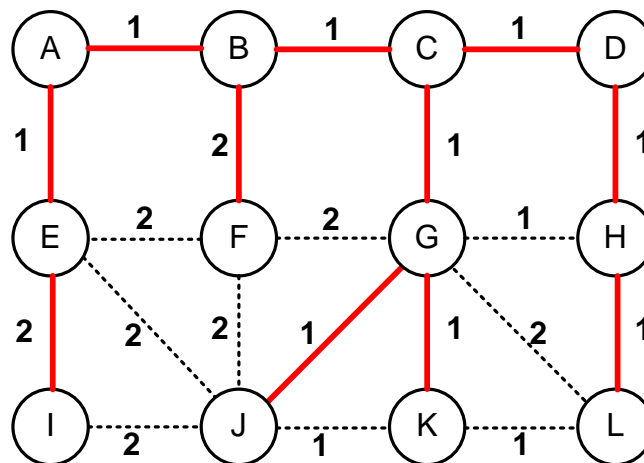
Nœud	A	B	C	D	E	F	G
Ordre:	2	4	3	5	1	6	7

Par Prim:

**Prim:**

Nœud	Distance	Parent	Connu?
A	∞ , 0	-	✓
B	∞ , 1	A	✓
C	∞ , 1	B	✓
D	∞ , 1	C	✓
E	∞ , 1	A	✓
F	∞ , 2	B	✓
G	∞ , 1	C	✓
H	∞ , 1	D	✓
I	∞ , 1	E	✓
J	∞ , 2	G	✓
K	∞ , 1	G	✓
L	∞ , 2, 1	G, H	✓

Par Kruskal:



Kruskal:

Arête	Poids	Retenue?
(A,B)	1	✓
(A,E)	1	✓
(B, C)	1	✓
(B, F)	2	✓
(C, D)	1	✓
(C, G)	1	✓
(D, H)	1	✓
(E, F)	2	
(E, I)	2	✓
(E, J)	2	
(F, G)	2	
(F, J)	2	
(G, H)	1	
(G, J)	1	✓
(G, K)	1	✓
(G, L)	2	
(H, L)	1	✓
(I, J)	2	
(J, K)	1	
(K, L)	1	