

IFT 2010 Structures de données

Professeure: Sylvie Hamel

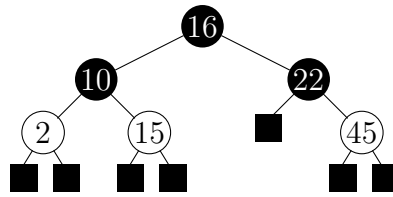
FINAL, Mardi le 20 avril, 2004

NOM:	QUESTION 1:	<input type="text"/>	20 pts
_____	QUESTION 2:	<input type="text"/>	5 pts
CODE PERMANENT:	QUESTION 3:	<input type="text"/>	20 pts
_____	QUESTION 4:	<input type="text"/>	10 pts
MOT (pour afficher les notes sur web):	QUESTION 5:	<input type="text"/>	10 pts
_____	QUESTION 6:	<input type="text"/>	10 pts
	QUESTION 7:	<input type="text"/>	25 pts
	TOTAL	<input type="text"/>	/100

Directives

- 1) Vérifiez que ce document comporte bien 13 pages.
- 2) Si vous voulez que j'affiche votre note sur le web, inscrire quelque chose sous MOT car je ne peux afficher sous votre nom ou code permanent.
- 3) Utilisez les versos des pages comme brouillons.
- 4) Examen à livre fermé i.e. aucune documentation permise.
- 5) Les règles concernant le plagiat seront strictement appliquées.

Question 1) [20 points] Soit l'arbre rouge-noir suivant :

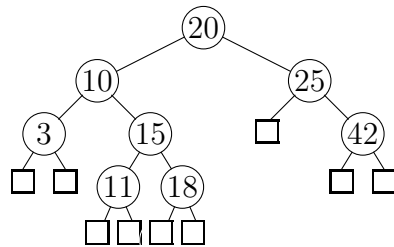


a) Dessinez un arbre (2,4) équivalent à cet arbre rouge-noir (expliquer l'équivalence)

b) L'insertion d'un élément de clé 17 dans l'arbre va créer un double-rouge. Décrivez comment rééquilibrer l'arbre rouge-noir et dessiner l'arbre après la restructuration.

- c) Est-ce que le double-rouge peut se propager vers la racine? Si oui, dans quel cas, si non, pourquoi?
- d) À quelle restructuration d'arbre (2,4) correspond la correction d'un double-rouge dans un arbre rouge-noir? (Dessiner la correspondance)
- e) Discuter de l'utilisation de la structure d'arbre rouge-noir versus arbre (2,4) (avantages et inconvénients de chaque méthode versus l'autre méthode).

Question 2) [5 points] Lorsqu'on fait la recherche d'une clé x dans un arbre binaire de recherche, on appelle la liste des noeuds visités, la **séquence de clés** de la recherche. Par exemple, lors de la recherche de 18 dans l'arbre suivant, la **séquence de clés** est 20, 10, 15, 18:



Supposons que nous avons un arbre binaire de recherche gardant en mémoire les clés 1, 2, 3, ..., 1000, et que l'on veuille faire la recherche de la clé 363. Expliquer pourquoi la séquence suivante NE PEUT PAS être la séquence de clés de votre recherche: 925, 202, 911, 240, 920, 245, 363.

Question 3) [20 points]

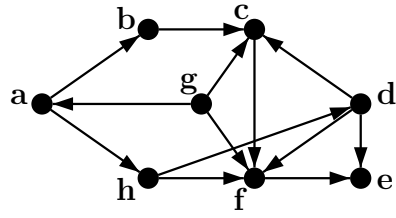
a) Qu'est-ce qu'un arbre AVL? (Définition + propriétés)

b) Démontrez que si un arbre AVL garde en mémoire n éléments, sa hauteur est en $\mathcal{O}(\log n)$

c) Décrivez comment supprimer un élément dans un arbre AVL (tous les cas).

d) Décrivez comment restructurer l'arbre après une suppression (en détails sur un exemple).

Question 4) [10 points] Trouver un ordre topologique pour les sommets du graphe qui suit. Dessinez les étapes de l'algorithme et le résultat final.



Question 5) [10 points] Supposons que vous avez un graphe connexe non-orienté G . Décrivez un algorithme qui détecte si G contient un cycle. Quel est la complexité en temps de votre algorithme, si on utilise une liste d'adjacences comme structure de données pour G .

Question 6) [10 points] Supposez qu'il y ait 8 îles sur un lac et qu'on veuille construire 7 ponts de sorte que toutes les îles soient reliées. Les plus petites distances en mètres entre les différentes îles sont données par la table suivante:

	1	2	3	4	5	6	7	8
1	-	240	210	340	280	200	345	120
2		-	265	175	215	180	185	155
3			-	260	115	350	435	195
4				-	160	330	295	230
5					-	360	400	170
6						-	175	205
7							-	305

Formulez ce problème comme un problème sur les graphes et trouvez quels ponts construire pour minimiser la longueur totale.

Question 7) [25 points] Supposons qu'un graphe non-orienté G soit représenté par la liste d'adjacences suivante:

Sommet	Sommets adjacents
1	2,3,4
2	1,3,4
3	1,2,4
4	1,2,3,6
5	6,7,8
6	4,5,7
7	5,6,8
8	5,7

a) Dessinez le graphe G

b) Donnez, dans l'ordre de visite, la séquence des sommets visités lors d'un parcours en profondeur de G (Depth-first search) à partir du sommet 1. Dessinez le graphe G avec les arêtes sélectionnées par l'algorithme en rouge (si vous avez) ou en trait pointillé (sinon).

c) Donnez, dans l'ordre de visite, la séquence des sommets visités lors d'un parcours en largeur de G (Breadth-first search) à partir du sommet 1. Dessinez le graphe G avec les arêtes sélectionnées par l'algorithme en rouge (si vous avez) ou en trait pointillé (sinon).

d) Donner la complexité en temps d'un des algorithmes de parcours de graphe et expliquer cette complexité. Vous pouvez supposer que la structure de données utilisée pour le graphe est une liste d'adjacences.

- e) Montrez que si le graphe G est un arbre, alors la complexité de l'algorithme de parcours en profondeur de G est linéaire en le nombre de sommets de G .