

IFT 2010 Structures de données

Professeure: Sylvie Hamel

EXAMEN FINAL, Mercredi le 27 avril, 2005

| | | | |
|--|-------------|----------------------|--------|
| NOM: | QUESTION 1: | <input type="text"/> | 24 pts |
| _____ | QUESTION 2: | <input type="text"/> | 16 pts |
| CODE PERMANENT: | QUESTION 3: | <input type="text"/> | 10 pts |
| _____ | QUESTION 4: | <input type="text"/> | 15 pts |
| MOT (pour afficher les notes sur web): | QUESTION 5: | <input type="text"/> | 5 pts |
| _____ | QUESTION 6: | <input type="text"/> | 20 pts |
| | QUESTION 7: | <input type="text"/> | 10 pts |
| | TOTAL | <input type="text"/> | /100 |

Directives

- 1) Vérifiez que ce document comporte bien 15 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) [24 points]

a) On ajoute à un arbre de recherche les clés suivantes 2, 15, 12, 8, 27, 24, 18 et 32, dans cet ordre. Dessinez la forme finale de l'arbre si l'arbre de recherche est

i) un arbre binaire de recherche

ii) un arbre AVL

iii) un arbre (2,4)

b) Maintenant, pour chacun des arbres obtenus en a), quels arbres obtenez-vous si vous enlevez les clés 15 et 18, dans cet ordre.

i) arbre binaire de recherche

ii) arbre AVL

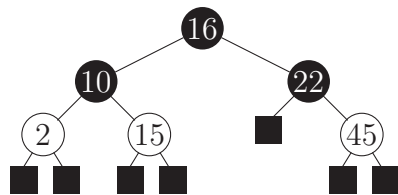
iii) arbre (2,4)

c) Donnez le pseudocode des reconstructions suivantes:

- i) Choisir une rotation dans un arbre AVL, la dessinez et donnez le pseudocode de la reconstruction.

ii) Donnez le pseudocode d'un fractionnement dans un arbre (2,4)

Question 2) [16 points] Soit l'arbre suivant:



a) Est-ce que cet arbre est un arbre rouge-noir? Si oui, pourquoi? Si non, pourquoi?

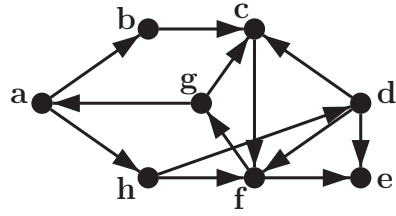
b) Existe-il des arbres rouges-noirs qui ne sont pas AVL? Si oui, donnez un exemple. Si non, expliquez pourquoi.

c) Qu'est-ce que le coloriage des noeuds et les propriétés sur ces couleurs apportent aux arbres rouges-noirs?

d) Donnez deux avantages de l'utilisation des arbres rouges-noirs versus les arbres (2,4).

Question 3) [10 points] Un graphe orienté G est fortement connexe si de chaque sommet de G , il est possible d'atteindre tous les autres sommets de G . Donner un algorithme en $\mathcal{O}(n + m)$ qui, étant donné un graphe orienté G , retourne VRAI si G est fortement connexe et FAUX, sinon. Expliquez la complexité.

Question 4) [15 points] Étant donné le graphe G suivant:



- a) Donnez une description brève de l'algorithme DFS (fouille ou recherche en profondeur) et un ordre de visite des sommets du graphe G possible lorsqu'on exécute DFS à partir du sommet a de G .

- b) Donnez une description brève de l'algorithme BFS (fouille ou recherche en largeur) et un ordre de visite des sommets du graphe G possible lorsqu'on exécute BFS à partir du sommet a de G .

c) Donnez et expliquez la complexité en espace et en temps d'un des deux algorithmes.

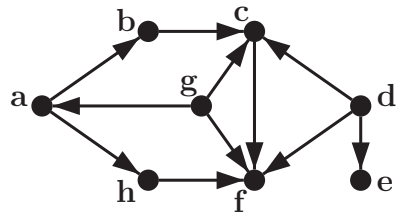
Question 5) [5 points] Considérez l'algorithme vorace suivant pour trouver le chemin de poids minimal entre le sommet "départ" et le sommet "arrivé", dans un graphe connexe:

1. Initialiser **chemin** avec le sommet "départ"
2. Initialiser **sommets visités** avec l'ensemble {"départ"}
3. Si "départ" = "but" alors retourner **chemin** et sortir
4. Trouver l'arête ("départ", v) de poids minimal telle que v soit adjacent à "départ" et v ne soit pas dans l'ensemble **sommets visités**
5. Ajouter v à **chemin**
6. Ajouter v à **sommets visités**
7. Mettre v dans "départ" et retourner à l'étape 3.

Est-ce que cet algorithme vorace trouve toujours le chemin de poids minimal entre les sommets "départ" et "arrivé"? Si oui, expliquez pourquoi, si non donnez un contre-exemple.

- c) Est-ce que l'algorithme fonctionne sur des graphes avec des arêtes de poids négatifs?
Expliquer pourquoi?

Question 7) [10 points] a) Trouver un ordre topologique pour les sommets du graphe qui suit.



b) Dans quel cas l'algorithme utilisé en a) ne trouve-t-il pas de solutions?