DÉPARTEMENT D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE

SIGLE DU COURS: IFT 2015 (E16)

NOM DU PROFESSEUR: Neil Stewart

TITRE DU COURS: Structures de Données

EXAMEN INTRA

Date	: Mercredi, 15 juin, 2016
Heure	: 9:00-11:00
Lieu	: Z-310

DIRECTIVES PÉDAGOGIQUES:

- Vous disposez de deux heures pour compléter cet examen.
- Documentation permise: une page 8" x 11", recto-verso.
- Mettez tout de suite votre nom et votre matricule dans la case (Figure 1).
- Répondez sur l'examen. Il y a deux pages brouillons (page 4 et page 9).
- L'espace alloué pour la réponse indique la longueur de la réponse cherchée.
- Il y a 10 questions, avec 100 points au total.

PLAGIAT. Constitue un plagiat:

- faire exécuter son travail par un autre
- utiliser, sans le mentionner, le travail d'autrui
- ÉCHANGER DES INFORMATIONS LORS D'UN EXAMEN
- falsifier des documents

Le plagiat est passible de sanctions allant jusqu'à l'exclusion du programme.

Figure 1: Mettez votre nom et matricule ici, ainsi que la mention "Prédoc I" si approprié.

1. Question 1 *(5 points)*

   Complexité

   Dans la définition de T_{best}, et dans la définition de $\Omega(\cdot)$, il y a l'idée de petitesse (minimisation ou borne inférieure).

   (a) Laquelle des deux choses (T_{best} ou $\Omega(\cdot)$) correspond à la minimisation?

   (b) Expliquez votre réponse: il s'agit de minimisation par rapport à quoi?

2. Question 2 *(5 points)*

   Exemple de meilleur cas

   Sans spécifier exactement l'ordre de clefs qui donnerait le résultat, expliquez comment les sous-listes doivent se diviser dans la méthode *Quicksort* pour avoir le meilleur cas.

3. Question 3 *(10 points)*

   Comparaison Espace

   Faisons la comparaison (espace) entre une implantation d'une liste par tableau, et une implantation par liste chaînée, avec des hypothèses particulières.

   Supposons que dans la liste chaînée, chaque noeud dans la liste prend deux pointeurs (chaque pointeur prend P octets), le premier pour indiquer le prochain noeud dans la liste chaînée, le deuxième pour pointer sur un objet "données" de classe *IElem* (E octets).

   Supposons aussi une implantation par tableau de dimension D, où chaque élément est un pointeur de P octets vers un objet "données" de classe *IElem* (E octets).

   S'il y a N éléments avec des données associées dans les deux cas, pour quelles valeurs de N le tableau sera meilleur?

4. Question 4 *(20 points)*

   Rebalancement AVL

   (a) Dans le cours nous avons vu le cas de rebalancement quand le noeud P avait des facteurs $d, d+2$, et un sous-arbre gauche $T1$ de hauteur d. L'enfant à droite de P a été noté S, et nous avons regardé le cas où l'insertion était à gauche de S: en effet, pour $d \geq 1$ l'enfant gauche de S a été noté R, R avait deux enfants $T2$ et $T3$, et le nouveau noeud a été inséré comme enfant de $T2$ ou $T3$. Le sous-arbre droit de S est $T4$ (de hauteur d).

   J'avais noté plus tard dans le cours que dans le cas $d = 0$ le sous-arbre $T1$ devient un seul noeud, R et ses enfants sont remplacés par le noeud inséré, et $T4$ aussi devient un seul noeud. Nous avons maintenant que $T2 = T3 = \Lambda$, et le noeud R est absent. Les deux rotations fonctionnent cependant comme d'habitude.

   - Faites le dessin de l'arbre dans le cas encore plus extrême de $d = -1$.
   - Nous avons toujours dans ce cas que $T2 = T3 = \Lambda$, et le noeud R est absent. Les sous-arbres $T1$ et $T4$ sont quoi maintenant?
   - Montrez les deux rotations nécessaires pour rebalancer l'arbre.

   (b) En remarquant que dans les deux cas, l'insertion à gauche de S et l'insertion à droite de S, la hauteur de l'arbre ne change pas apres le ou les rotation(s), quelqu'un arrive à la conclusion que nous n'avons plus besoin de mémoire externe (les disques, par exemple), parce que les arbres AVL en mémoire interne ne changent jamais de hauteur, indépendemment du nombre de clefs que nous insérons dans l'arbre. Et pourtant, il me semble qu'il devrait avoir un hic quelque part dans cette idée!

   Comment la hauteur de l'arbre peut devenir plus grande?

Page brouillon: cette page ne sera pas prise en compte lors de la correction.

5. *Question 5* (10 points)
Listes généralisées

(a) Faites une dessin d'une liste pure avec entête, avec une dizaine de noeuds, et avec au moins une sous-liste. Aussi, cette sous-liste doit avoir un noeud qui a lui-même une sous-liste.

(b) Nous avons vu une représentation d'arbre basée sur l'idée que chaque noeud se souvient de son "sibling" le plus vieux après lui, et de son enfant le plus vieux. Donnez l'arbre qui correspond à liste pure que vous avez dessinée dans la partie 5a. (La racine correspond à l'entête de liste.)

6. *Question 6* (10 points)

   Les monceaux

   Emmagasinons les clefs d'un monceau dans une structure linéaire, avec la première clef dans la case 1, comme nous avons fait dans le cours.

   (a) Dessinez l'arbre "Complete" qui a les 12 clefs suivantes: $2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37$.

   (b) Dessinez l'arbre "Complete" avec 11 clefs, après l'opération *delete_min*. Expliquez brièvement les échanges qui ont été faites pour y arriver.

7. *Question 7* (10 points)

   Retrait d'un noeud

   Nous avons vu dans le cours la méthode standard utilisée pour retirer un noeud d'un arbre binaire in-order (GRD), méthode basée sur une échange avec le prédecesseur. Pour trouver le noeud prédecesseur, il faut suivre un certain chemin. Dessinez l'arbre avec N noeuds qui maximise la longueur de ce chemin.

8. *Question 8* (10 points)

Estimé de la solution d'une récurrence

Pour estimer la hauteur d'un arbre "aléatoire", nous avons dû résoudre une récurrence de la forme

$$N \cdot D(N) = 2\left[\sum_{j=0}^{N-1} D(j)\right] + f(N), \qquad N \geq 2.$$

Quelle était la première grande étape de cette preuve, c'est-a-dire l'astuce principale pour résoudre la récurrence? Cela nous a donné quoi pour $D(N)/(N+1)$?

9. *Question 9* (10 points)

   Remarque dans le Devoir 2.

   Il a été dit dans l'énoncé du Devoir 2 qu'un noeud x est l'ancêtre d'un noeud y si et seulement si x précède y dans le parcours "pre-order" et x succède y dans le parcours "post-order". Donnez la preuve de cet énoncé.

Question 10 (10 points)

Adressage dispersé (Hashing)

(a) Dans la méthode de Listes Externes, nous choisissons dans la pratique une valeur de M (nombre de cases) en fonction de N, la valeur attendue du nombre de clefs. Le but de ce choix est de garder quelle quantité (quelle fonction de N et de M) petite?

(b) Quel serait l'intéret de remplacer les listes linéaires, dans la méthode de Listes Externes, par un arbre AVL?

Page brouillon: cette page ne sera pas prise en compte lors de la correction.

Fin de l'examen.

Neil Stewart