DÉPARTEMENT D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE

SIGLE DU COURS: IFT 2015 (E15)

NOM DU PROFESSEUR: Neil Stewart

TITRE DU COURS: Structures de Données

EXAMEN INTRA

Date	: Mercredi, 17 juin, 2015
Heure	: 10:00-12:00
Lieu	: AA-1355

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 5 et page 11).
- 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 *(15 points)*

   Arbres binaires de recherche

   (a) En vous servant simplement de trois lettres majuscules, donnez la définition de l'ordre (au sens de "ordre de parcours") utilisé normalement dans les arbres binaires de recherche.

   (b) Expliquez comment créer un arbre qui respecte cet ordre, en supposant disponibles des clefs distinctes.

   (c) Dans le cas de clefs qui peuvent être identiques (donc pas nécessairement distinctes), on peut insister que toutes les clefs dans le sous-arbre gauche d'un noeud sont *strictement* inférieures à la clef dans le noeud. Cela revient à dire que les clefs égales sont toujours mises dans le sous-arbre droit. Montrer le résultat d'utiliser l'algorithme de "création d'arbre" dans le cas de clefs F, E, D, F, E, A, E. (Les clefs sont ordonnées par l'ordre lexicographique habituel, et elles sont traitées dans l'ordre indiqué.) Expliquez aussi l'avantage d'insister que toutes les clefs dans le sous-arbre gauche d'un noeud sont *strictement* inférieures à la clef dans le noeud.

   Explication de l'avantage mentionné:

2. Question 2 (10 points)

   À la page 118 du livre de Weiss, une stratégie de *retrait paresseux* est suggérée pour les arbres, utile si les clefs retirées risquent d'être réinsérées. Plutôt que de retirer une clef, nous allons simplement marquer le noeud comme *retiré* (R). Pour commencer, si jamais la clef est réinsérée, il ne sera pas nécessaire d'allouer de l'espace de nouveau. Mais, en plus, si le nombre de noeuds qui sont toujours présents dans l'arbre est à peu près le même que le nombre de noeuds marqués "R", alors l'augmentation de la profondeur de l'arbre ne devrait pas être plus qu'une petite constante. Weiss vous demande pourquoi.

   Pourquoi?

3. Question 3 *(10 points)*

   Implantation du tri topologique

   Dans le Devoir 2 vous avez implanté le tri topologique. Quel était le rôle de la classe *Element* dans cette implantation?

   (Note: Les étudiants PhD qui font le Prédoc I ne sont pas obligés de faire les devoirs. Ces étudiants peuvent répondre simplement ici "Prédoc I".)

4. Question 4 *(5 points)*

   Nous avons remarqué une façon alternative de représenter un arbre (pas nécessairement binaire), soit la représentation "puînée", où chaque noeud se souvient de son enfant le plus vieux, et de son "sibling" le plus vieux après lui. Si je conçois cela comme un exemple de liste généralisée, il s'agit de quelle sous-classe de liste généralisée?

5. Question 5 *(10 points)*

   (a) Donnez une preuve simple que la création d'un monceau est $O(N \log N)$.

   (b) En fait, il y a un meilleur algorithme.

      i. Quel est le nom de cet algorithme?

      ii. Quelle est la complexité de cet algorithme?

      iii. Expliquez brièvement le fonctionnement de cet algorithme.

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

6. Question 6 *(10 points)*

   Monceau et sac

   (a) Dans le TAD *Queue de priorité* il y a l'opération *insérer*. En supposant que vous utilisez un monceau, expliquez brièvement comment cette opération se fait, et donnez la complexité (le coût) de l'opération.

   (b) Dans le TAD *sac* (anglais "bag") (exemple: simulation des particules), il y a l'opération *retirer(x)*. En supposant que vous utilisez un tableau linéaire, et en supposons x déjà trouvé, expliquez brièvement comment cette opération se fait, et donnez la complexité (le coût) de l'opération.

7. Question 7 *(10 points)*

   Supposons que nous ayons une référence sur un noeud P de type $NoeudAVL$, où

$$\begin{aligned} P.gauche &= T1 \\ P.droit &= S \end{aligned}$$

   et S (aussi de type $NoeudAVL$) a

$$\begin{aligned} S.gauche &= T2 \\ S.droit &= T3. \end{aligned}$$

   Supposons aussi que nous avons inséré un nouveau noeud dans $T3$ (c'est l'exemple du cours).

   Si je dois maintenant faire une rotation simple pour amener S à la racine, je commence par prendre note de $P.droit$ en faisant l'affectation

$$Temp \leftarrow P.droit$$

   et je fais deux autres affectations (lesquelles?):

   (a)

   (b)

   avant de retourner $Temp$ comme racine.

   Donnez les dessins (avant et après la rotation), et donnez les affectations (a) et (b). *Indice:* Les deux affectations impliquent $Temp.xxx$.

8. Question 8 *(10 points)*

   Weiss donne une preuve similaire à celle que nous avons donnée dans le cas de la profondeur d'un arbre de recherche binaire moyen. (Cela sert aussi pour l'analyse de *Quicksort*.)

   Dans sa notation nous avons

   $$NT(N) = (N+1)T(N-1) + 2cN$$

   où c est une constante. (Aussi, prenons $T(1) = 0$.)

   En utilisant le principe de "telescoping", et le fait que les nombres harmoniques H_N sont $\Theta(N)$, démontrez que $T(N) = O(N \log N)$.

9. Question 9 *(15 points)*

   Complexité

   Notons que $T_{best} \leq T_{worst}$. Expliquez dans les deux cas:

   (a) Est-il possible que T_{best} soit $O(N^3)$, en mêmes temps que T_{worst} soit $O(N \lg N)$?

   (b) Est-il possible que T_{best} soit $\Omega(N^2)$, en mêmes temps que T_{worst} soit $O(N \lg N)$?

   (c) En utilisant la définition de Ω, montrez que si une fonction est $\Omega(\log N)$, alors elle est aussi $\Omega(\frac{1}{N} \log N)$.

10. Question 10 *(5 points)*

    Expliquez brièvement les deux approches principales de resolution de collision dans la méthode de Hashing.

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

Fin de l'examen.

Neil Stewart