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 FINAL

Date : Lundi, 27 juillet, 2015
Heure : 13:00-16:00
Lieu : AA-1355

DIRECTIVES PÉDAGOGIQUES:

- Vous disposez de trois 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 6 et page 10).
- 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.

1. Question 1 *(10 points)*

   Arbres 2-3

   À titre d'exemple, nous avons discuté en détail de la façon d'insérer et de retirer des clefs dans un arbre 2-3. Dans les deux cas, il a fallu de temps en temps remonter dans l'arbre, et de modifier le positionnement des clefs.

   (a) Donnez un exemple simple où il n'est pas possible d'insérer la nouvelle clef au plus bas niveau d'un arbre 2-3 (expliquez clairement pourquoi il n'est pas possible), mais où il est possible d'insérer la clef en remontant à l'avant-dernier niveau.

   (b) Il est facile de dire "remonter dans l'arbre", mais comment faire cela dans la pratique? Un arbre normalement n'a que des pointeurs qui s'en vont dans le sens "racine vers feuilles". Expliquez.

2. Question 2 *(15 points)*

(a) *Algorithme de Prim (ou algorithme de Dijkstra)*

Dans l'algorithme de Prim, nous faisons une boucle extérieure sur les N noeuds du graphe, ce qui fait que nous avons N étapes à réaliser.

À chaque étape il a fallu trouver une certaine distance minimale, et cette partie du processus exigeait en moyen $N/2$ étapes, en supposant que nous n'avons pas utilisé un monceau pour trouver les minimums.

Aussi, à chaque étape il y a une boucle interne sur les voisins du noeud que nous traitons, et en principe il pourrait avoir jusqu'à $N-1$ voisins pour chaque noeud traité. Et pourtant, nous ne nous sommes pas contentés de donner un estimé de $\Theta(N^2)$ pour le coût de cette deuxième partie du processus: nous avons pu donner un estimé qui est parfois plus utile, en étant un peu plus subtil dans notre analyse. Expliquez.

(b) *Parcours avec ficelle*

Une idée similaire est utilisée dans l'analyse du parcours d'un graphe en utilisant des ficelles. Si en allant de gauche à droite nous suivons un lien normal à telle ou telle étape, nous devons ensuite descendre à gauche jusqu'au moment où nous trouvons une ficelle à gauche. Pour chaque noeud, en principe on pourrait descendre $\Theta(N)$ noeuds, et pourtant, le coût total du parcours n'est pas $\Theta(N^2)$, c'est plutôt $\Theta(N)$. Expliquez.

3. Question 3 *(5 points)*

   Pierres tombales

   En utilisant l'addressage ouvert pour le "hashing", on pourrait utiliser des sentinels ("pierre tombales") pour indiquer les clefs retirées. L'algorithme d'insertion peut se servir de cellules avec pierre tombale pour insérer une clef. En arrivant sur une pierre tombale, dans quelles circonstances l'algorithme peut insérer directement la nouvelle clef, sans faire autre chose avant?

4. Question 4 *(10 points)*

   Supposons un arbre rouge-noir, avec cinq noeuds, et avec la clef 4 à la racine. L'enfant droit de la racine contient la clef 5, tandis que l'enfant gauche (rouge) de la racine contient la clef 2. Enfin, les enfants du noeud qui contient la clef 2 contiennent respectivement les clefs 1 et 3.

   Dessinez l'arbre rouge-noir (en indiquant les couleurs des noeuds), dessinez un arbre 2-4 équivalent, et dessinez un "1-2-3 Deterministic Skip-List" équivalent.

5. Question 5 *(10 points)*

Dessinez le résultat de faire les opérations suivantes, à partir d'un arbre vide, pour un SplayTree. Indiquez les sous-étapes au besoin.

(a) Insérez les clefs 1, 2, 3, 4, dans l'ordre.

(b) Accéder à la clef 1.

(c) Ensuite, dans une opération séparée, retirer la clef 1.

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

6. Question 6 *(10 points)*

Dans la Figure 2 nous voyons un exemple de SkipList randomisé. Les niveaux dans l'exemple au stade actuel sont 0, 1, 2 et 3.

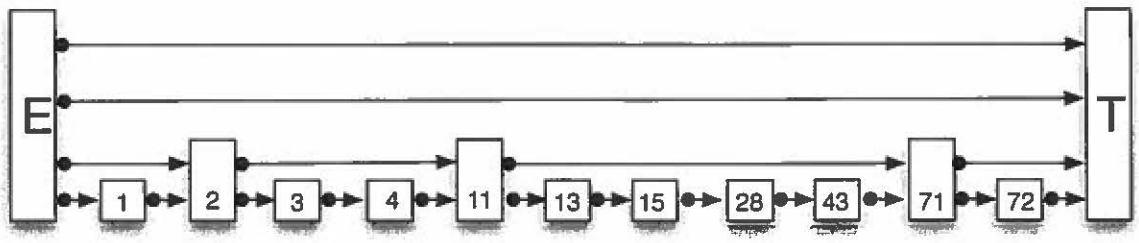

Figure 2: Skiplist

Ensuite, nous voulons insérer la clef 17. Pour ce faire, nous faisons appel à la procédure

$$\textbf{pour}(nivmax = 0; random() == 0; nivmax{+}{+});$$

qui par hasard nous retourne trois fois 0, suivi d'un 1, c'est-à-dire 0, 0, 0, 1.

Dessinez le SkipList après l'insertion de la clef, et expliquez comment le résultat serait différent si la procédure nous avait donné tout de suite un 1 (aucun 0).

7. Question 7 *(10 points)*

   Hachage double

   Nous avons utilisé une fonction $h(x)$ pour trouver une case pour mettre la clef, et une deuxième fonction $p(x)$ pour faire le ciblage en cas de collision. Pourquoi cela n'aurait pas de sens à utiliser deux fois la fonction h, pour éviter le calcul de la deuxième fonction p? (Il suffirait juste de prendre 1 à la place de 0 dans le cas $h(x) = 0$.)

8. Question 8 *(10 points)*

   Parcours d'un graphe

   Soit un graphe non-orienté (V, E), où $V = \{A, B, C, D\}$, qui a la forme d'un cycle simple: les listes d'adjacence sont $A : B, D$; $B : C, A$; $C : D, B$; $D : A, C$.
   Montrez le fonctionnement de l'algorithme pour un parcours en profondeur.

9. Question 9 *(10 points)*

Arbres B

Pour les arbres B, la valeur du paramètre M est normalement déterminée par la taille d'une page (pour les données externes, cela serait déterminée par la taille d'un bloc sur disque). Sujet à la contrainte que les clefs et pointeurs associés peuvent s'enregistrer sur une page, nous voulons normalement prendre M aussi grand que possible. Expliquez pourquoi, et expliquez pourquoi nous insistons que le nombre de pointeurs est normalement au moins $\lceil M/2 \rceil$.

10. Question 10 *(10 points)*

SplayTree: retrait

Dans les Splay tree, pour retirer un noeud, nous commençons par accéder au noeud à retirer, ce qui envoit le noeud à retirer à la racine. Cette racine a maintenant les sous-arbres T_L et T_R. À la prochaine étape, nous trouvons l'élément le plus petit dans T_R, et nous faisons quelque chose avec cet élément et l'arbre que nous avons obtenu.

(a) Qu'est-ce que nous faisons exactement avec cet élément? Expliquez avec un petit dessin.

(b) Mais qu'est-ce que nous faisons si T_R est vide (et l'élément le plus petit n'existe donc pas)?

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

Fin de l'examen.

Neil Stewart