Nom :	**Prénom :**	*page 1*

UPMC - UFR 919 - Master d'informatique - M1

Bases de Données Réparties- MI034 – 1er examen réparti du 24 février 2014

Version CORRIGEE

ELEMENTS DE SOLUTION

Exercice 1 : Arbres B+ — 7 pts

Tous les arbres sont d'ordre 2 (i.e., de 2 à 4 valeurs par nœud sauf la racine qui a de 1 à 4 valeurs). Utiliser la syntaxe suivante pour représenter un nœud de l'arbre : *N (v_1, v_2, …)* où *N* est le nom du nœud et les v_i sont les valeurs. Quand la feuille F déborde, on garde les 3 plus petites valeurs dans F, les 2 plus grandes valeurs vont dans une nouvelle feuille.

Soit l'arbre A1 composé d'une racine **N_1**(90), et 2 feuilles **F_1**(0,1) et **F_2**(99,100).

1) On insère successivement les nombres entiers consécutifs croissants {2, 3, …, 10} dans les feuilles. Que contient N_1 après la dernière insertion ?

N1(3, 6, 9, 90)

2) Pourquoi n'obtient-on pas l'arbre A2 suivant ?
Une racine **N_1**(4, 8, 90) et les feuilles : **F_1**(0, 1, 2, 3), **F_2**(4, 5, 6, 7), **F_3**(8, 9, 10) et **F_4**(99, 100) ?

Parce qu'un éclatement crée des feuilles avec 3 et 2 valeurs (et non 4 valeurs) et que les nombre insérés étant croissants, on n'insère jamais dans la feuille avec 3 valeurs qui résulte de l'éclatement.

3) Lister, dans l'ordre, les entiers à insérer successivement **dans A1** pour obtenir l'arbre A3 suivant :
Une racine **N_1**(4, 90) et les feuilles **F_1**(0, 1, 2, 3), **F_2**(4, 5, 6, 7) et **F_3**(90, 91, 99, 100).

Insérer le 3 après 2,4, 5

Autre réponse : insérer 2 après 3,4,5

4) A partir de A1, on insère successivement les entiers consécutifs {2, 3, …, n} de telle sorte que l'insertion de la valeur *n* provoque le débordement de N1 (pour la première fois) et l'ajout d'un niveau.
Que vaut n ? Décrire la racine de l'arbre obtenu et ses nœuds intermédiaires. Rmq : on ne demande **pas** de préciser toutes les feuilles.

n = 13

Racine R(9)

N1(3,6) N2(12, 90)

5) A partir de A1, on insère successivement les entiers consécutifs {2, 3, …., 98}. On obtient l'arbre A4 après la dernière insertion. L'arbre A4 a 101 valeurs dans ses feuilles.

a) Que contiennent les 2 feuilles situées les plus à droite ?

….. (96,97,98) (99,100)

b) Combien A4 a –t-il de feuilles ?

34

c) Quel est le nombre de valeurs par nœuds, dans un nœud parent d'une feuille ?

2 valeurs par nœud sauf le nœud le plus à droite qui peut avoir de 2 à 4 valeurs.

d) (question bonus) Combien A4 a –t-il de niveaux (niveau de la racine inclus) ?

4 niveaux (profondeur 3)

6) On traite la suppression d'une valeur en redistribuant, si possible, avec la feuille de gauche puis la feuille de droite (si leur père est le même). On considère l'arbre A5 suivant :

- la racine est **R** (15),
- le niveau intermédiaire contient 2 nœuds **N1** (5, 10) et **N2** (20, 25),
- les feuilles sont **F1** (2, 4), **F2** (6, 8), **F3** (13, 14), **F4** (15, 16, 17), **F5** (20, 22) et **F6** (28, 30).

a) On supprime 22 dans A5, quels nœuds ont été supprimés ou modifiés (et que contiennent-ils) ?

F4(15,16) F5(17,20) et ajuster N2(17,25)

b) On supprime 14 dans A5, quels nœuds ont été supprimés ou modifiés (et que contiennent-ils) ?

Suppr F3, N1 et R

N2 (5,15,20,25)

F2(6,8,13)

Exercice 2 : Table de hachage extensible 7 pts

Dans toutes les tables de hachage considérées, un paquet ne peut pas contenir plus de 4 valeurs. Lors d'une suppression, si un paquet devient vide, on tente de le fusionner avec un autre paquet, si ce n'est pas possible, il reste vide.
Utiliser la syntaxe suivante :

L (v_i, ...) PL=n avec L le nom (A, B, ...) du paquet, *n* la profondeur locale et les valeurs v_i.
Rmq : utiliser la lettre suivante de l'alphabet pour nommer un nouveau paquet.
R[L, L', ...] pour le répertoire avec L, L', ... les noms des paquets.
On peut aussi préciser le contenu d'une case particulière avec R[i]= L (avec R[0] étant la 1ère case).

Une table de hachage T1 contient *seulement* les 5 paquets suivants:

A (8)
B (9, 29)
C (10, 22, 42)
D (11, 27)
E (23, 39)

1) Structure de T1.

a) Est-ce que le répertoire est tel que R[0] = A et R[4] = E ?

R[0] = A ? Oui

R[4]=E ? Non

b) Pour chaque paquet, quelle est sa profondeur locale ?

A (8) PL=2

B (9, 29) PL=2

C (10, 22, 42) PL=2

D (11, 27) PL=3

E (23,39) PL=3

c) Que contient le répertoire? Préciser le nom du paquet référencé dans chaque case.

Taille=8

R[A, B, C, D, A, B, C, E]

2) Dans T1, on insère successivement 6 puis 18. Quels sont les paquets créés et/ou modifiés et leur contenu ? Préciser aussi les cases modifiées du répertoire.

6 mod 8 = 6, insérer 6 dans C(6,10,22,42)

18 mod 8 = 2 , insérer 18 dans C débordement

Création de F

Répertoire R[6]=F R[A, B, C, D, A, B, **F**, E]

et répartition des valeurs dans C(10,18,42) F(6,22)

3) Dans T1, on insère successivement 3 valeurs dans le paquet E. Est-ce que cela a pour effet de doubler la taille du répertoire ?

Oui, car E a une PL=3 identique à la PG donc si E déborde, il faut doubler le répertoire

4) Dans T1, on supprime 9 puis 29. Quels sont les paquets supprimés et/ou modifiés et leur contenu ?

Pas de fusion avec D car D n'a pas la même profondeur locale que B

B reste vide avec PL=2

5) Dans T1, on supprime 8. Quels sont les paquets supprimés et/ou modifiés et leur contenu ?

Suppr 8 dans A qui devient vide. **Pas** de fusion car PL(A) < PG

~~Fusion avec C et réduire la profondeur locale. C[0] = C C(10, 22, 42) PL = 1~~

6) On a un million (10^6) de valeurs à indexer. Quelle sera la taille minimale du répertoire ?

2^{18}

7) Question bonus :

Exercice 3 : Optimisation de requêtes 6 pts

Une base contient

Restau (num, nom, étoile, ville, tel) // *num* est un numéro de restau.

Avis (pseudo, num, date, note, texte) // l'utilisateur *pseudo* a attribué une *note* au restau *num*.

Les étoiles vont de 1 à 5, les notes vont de 0 à 20 inclus. Il y a 100 villes. On suppose la distribution uniforme des valeurs d'un attribut. Les attributs sont indépendants.
On a :

card(Avis)=100 000 et card(Restau) = 10 000

R1 :
```
select *
from Avis a, Restau r
where a.num = r.num
and a.note between 12 and 17
and r.étoile = 3 and r.ville='Paris'
```

Pour simplifier la notation les prédicats sont appelés pi comme suit :

p1 : a.num = r.num *p2* : a.note between 12 and 17
p3 : r.étoile = 3 *p4* : r.ville='Paris'

1) Quels sont les facteurs de sélectivités des sélections exprimées dans R1

SF(σ_{p2} (Avis))= 6/21 = 0,29

SF(σ_{p3} (Restau)) = 1/5 = 0,2

SF(σ_{p4} (Restau)) = 1/100 = 0,01

2) Détailler le calcul de card(R1)

Card(Restau jointure Avis) = card(Avis) = 100 000

Card(R1) = 100 000 * 0,29 * 0,2 * 0,01 = 58

3) On considère les expressions suivantes pour évaluer R1.

E1 : $\sigma_{p4}(\sigma_{p3}(\sigma_{p2}(\text{Avis} \bowtie_{p1} \text{Restau})))$

E2 : $[\sigma_{p2}(\text{Avis})] \bowtie_{p1} [\sigma_{p4}(\sigma_{p3}(\text{Restau}))]$

E3 : $[\sigma_{p4}(\sigma_{p3}(\text{Restau}))] \bowtie_{p1} [\sigma_{p2}(\text{Avis})]$

E4 : $\sigma_{p4}(\sigma_{p3}[\sigma_{p2}(\text{Avis}) \bowtie_{p1} \text{Restau}])$

E5 : $\sigma_{p2}([\sigma_{p4}(\sigma_{p3}(\text{Restau}))] \bowtie_{p1} \text{Avis})$

On rappelle les formules de coût vues en TD (ex. 2 pages 9) :

coût(σ_p(R)) = card(σ_p(R)) si l'attribut du prédicat p est indexé
= card(R) sinon.

coût(R $\bowtie_a$ S) = card(R) si S.a est indexé
= card(R) × card(S) sinon

Le parcours séquentiel de R coûte card(R). On *néglige* le coût des opérateurs traités en pipeline et le coût d'écrire les résultats intermédiaires.

a) Il n'y a aucun index. Quelle est, parmi E1 à E5, l'expression de moindre coût ? Il n'est pas nécessaire de préciser le coût.

Faire d'abord les sélections donc E2 ou E3

b) Il y a un index sur Restau.num et un autre sur Avis.num. Quelle est, parmi E1 à E5, l'expression de moindre coût ?

Utiliser l'index pour la jointure. Commencer par lire la relation qui est la plus filtrée par sélection : E5

c) Il y a un index sur Avis.note et un autre sur Avis.num. Quelle est, parmi E1 à E5, l'expression de moindre coût ?

Pas intéressant d'utiliser Avis.note car cela empêche ensuite d'utiliser Avis.num

Donc **E5** a le coût mini.

d) Tous les attributs ont un index. Quelle est l'expression E' de moindre coût et quel est son *coût* ? Quels sont les index utilisés ?

E'= σ_{p2} ([σ_{p3} (σ_{p4} (Restau))] $\bowtie_{p1}$ Avis) Index utilisés Restau.ville et Avis.num

Autre réponse acceptée : utilisation combinée des index sur ville et étoile (avec une intersection comme dans oracle avec l'opérateur BITMAP AND ou AND EQUAL)

e) Quel(s) index faut-il avoir pour que E4 devienne l'expression de moindre coût parmi E1 à E5 ?

Index sur Avis.note et Restau.num