

M1 informatique - Optimisation des bases de données

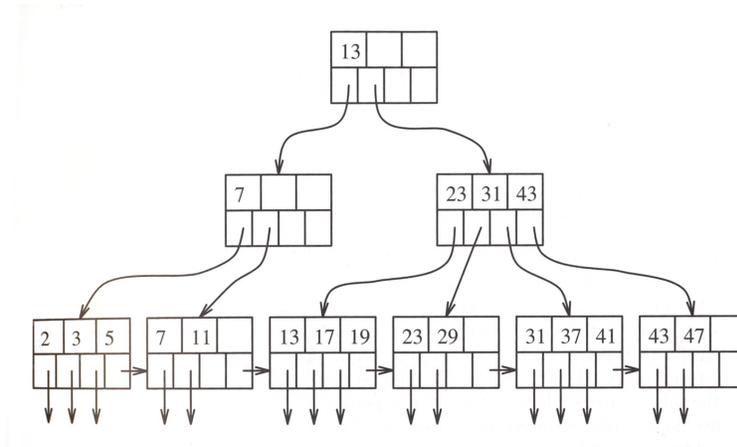
TD 2 : indexation

1 Structure de B+TREE

Combien de B+TREE sont possibles si $n=3$ et la table indexée contient 6 enregistrements? Même question pour 10 enregistrements?

2 Mise à jour

Considérons le B+TREE donné ci-dessous.



Décrire les évolutions de l'arbre pour les mises à jour successives suivantes :

- insertion d'un enregistrement de clé égale à 1,
- insertion des enregistrements de clé égale respectivement à 14, 15, 16,
- suppression des enregistrements de clé égale ou supérieure à 23.

3 Indexation sur clé primaire

Considérons la table purchase définie comme suit :

```
CREATE TABLE purchase(  customer_id UNSIGNED SMALLINT
                          NOT NULL AUTO_INCREMENT,
                          store_name CHAR(15) NOT NULL,
                          purchased_item CHAR(10) NOT NULL,
                          purchase_date DATETIME NOT NULL,
                          PRIMARY KEY(customer_id));
```

Supposons que la table purchase soit ensuite remplie avec 10 000 enregistrements.

Si on indexe les enregistrements de la table sur l'attribut customer_id avec un B+TREE, sachant que les blocs ont une taille de 4 Koctets, les pointeurs une taille de 8 octets et les clés sont des integers sur 2 octets :

1. quel est l'ordre de ce B+TREE (on suppose qu'il n'y a pas de header dans le bloc, et que les données sont stockées dans un fichier séquentiel, trié sur la clé)?
2. Combien de blocs sont nécessaires au stockage de ce B+TREE si chaque feuille contient le maximum de clés? Même question si les feuilles du B+TREE sont pleines à 75% en moyenne? Même question si les feuilles du B+TREE sont pleines à 50%?
3. Quel est le nombre de lectures disque pour répondre à la requête suivante (on suppose que rien n'est en mémoire initialement)?

```
SELECT  *
FROM    purchase
WHERE   customer_id=5000;
```

4. Reprendre les questions précédentes en supposant que le nombre d'enregistrements de la table est passé à 65 536. Reprendre les questions précédentes si on considère que l'index est creux.