Chapître XI.
B-arbres

1. Structures de données avancées

-) tas binomial

-) tas de Fibonacci

-) structure de données d'ensembles disjoints

-) <u>B-arbres</u>

2. Arbres de recherche équilibrés

Les B-arbres sont des arbres de recherche équilibrés conçus pour être efficaces sur des disques magnétiques ou d'autres unités de stockage secondaires à accès direct.

Le différence majeure entre les **B-arbres** et les autres arbres de recherche équilibrés comme les **arbres AVL** ou les **arbres rouge et noir** réside dans le fait que **les nœuds des B-arbres peuvent avoir de nombreux fils**, jusqu'à plusieurs milliers.

Le "facteur de branchement" d'un B-arbre peut être très grand, bien qu'**il soit généralement déterminé par les caractéristiques de l'unité de disque utilisée.**

3. Exemple

4. Définition

Un **B-arbre** T est un arbre enraciné possédant les propriétés suivantes.

1. Chaque nœud x contient les champs ci-dessous:

a. $n[x]$, le nombre de clés conservées par le nœud x,

b. les $n[x]$ clés elles-mêmes, stockées par ordre croissant: $cle_1[x] \leq cle_2[x] \leq \cdots \leq cle_{n[x]}[x]$, et

c. $feuille[x]$, une valeur booléenne qui vaut $VRAI$ si x est une feuille et $FAUX$ si x est un nœud interne.

2. Si x est un nœud interne, il contient également $n[x]+1$ pointeurs $c_1[x], c_2[x], \ldots, c_{n[x]+1}[x]$ sur ses fils. Les feuilles n'ont aucun fils, et leurs champs c_i ne sont donc pas définis.
3. Les clés $cle_i[x]$ déterminent les intervalles de clés stockés dans chaque sous-arbres: si k_i est une clé quelconque stockée dans le sous-arbre de racine $c_i[x]$, alors

$$k_1 \leq cle_1[x] \leq k_2 \leq \cdots cle_{n[x]}[x] \leq k_{n[x]+1}.$$

4. Toutes les feuilles ont la même profondeur, qui est égale à h, la hauteur de l'arbre.

5. Il existe des bornes inférieures et supérieures sur le nombre de clés pouvant être contenues par un nœud. Ces bornes peuvent être exprimées en fonction d'un entier fixé $t \geq 2$, appelé le **degré minimum** du B-arbre:

a. Tout nœud autre que la racine doit contenir au moins $t-1$ clés. Tout nœud interne autre que la racine possède donc au moins t fils. Si l'arbre n'est pas vide, la racine doit posséder au moins une clé.

b. Tout nœud peut contenir au plus $2t-1$ clés. Un nœud interne peut donc posséder au plus $2t$ fils. On dit qu'un nœud est **complet** s'il contient exactement $2t-1$ clés.

5. Arbres 2-3-4

En 1970, J.E. HOPCROFT inventa les **arbres 2-3** dans lequels tout nœud interne possède soit deux, soit trois fils.

Les **arbres 2-3** sont des précurseurs des **arbres 2-3-4** et des **B-arbres**, qui ont été introduits en 1972.

Le **B-arbre** le plus simple est celui de degré minimum $t = 2$. Tout nœud interne possède alors 2,3, ou 4 fils, et on se trouve en présence d'un **arbre 2-3-4**.

Toutefois, en pratique, on utilise des valeurs de t beaucoup plus grandes.

6. La hauteur d'un B-arbre

Le nombre d'**accès au disque** nécessaires pour la plupart des opérations sur un B-arbre est proportionnel à la hauteur du B-arbre.

Théorème. *Si $n \geq 1$, alors pour un B-arbre T à n clés, de hauteur h et de degré minimum $t \geq 2$:*

$$h \leq \log_t \frac{n+1}{2} = \frac{\log_2 \frac{n+1}{2}}{\log_2 t}.$$

Démonstration: Si un B-arbre a une hauteur h, le nombre de ses nœuds est minimisé quand la racine contient une seule clé et que tous les autres nœuds contiennent $t-1$ clés.

Dans ce cas, il existe:

1 nœud de profondeur 0,
2 nœuds de profondeur 1,
$2t$ nœuds à la profondeur 2,
$2t^2$ nœuds à la profondeur 3,
...
$2t^i$ nœuds à la profondeur $i + 1$,
... et ainsi de suite,
$2t^{h-1}$ nœuds à la profondeur h.

Puisque chaque nœud excepté la racine contient $t-1$ clés, le nombre n des clés satisfait l'inégalité

$$\begin{aligned} n &\geq 1 + (t-1)\sum_{i=1}^{h} 2t^{i-1} \\ &= 1 + 2(t-1)\Big(\frac{t^h - 1}{t-1}\Big) \\ &= 2t^h - 1 \end{aligned}$$

ce qui implique

$$h \leq \log_t \frac{n+1}{2}.$$

$\square$

La hauteur de l'arbre augmente proportionellement à $\log n$ dans le cas des arbres rouge et noir et aussi dans le cas des B-arbres, car ce sont des arbres de recherche équilibrés.

Dans les cas des B-arbres, la base du logarithme peut être bien supérieure.

Les B-arbres économisent un facteur d'environ $\log t$ par rapport aux arbres rouge et noir, en hauteur, et en nombre de nœuds examinés, pour la plupart des opérations sur les arbres.

7. Les structures de données sur un espace de stockage secondaire

Nous avons besoin d'une petite révision de système parce qu'on utilise les B-arbres en particulier quand il s'agit de **minimiser les opérations d'entrées/sorties sur disque**.

Pour un grand B-arbre stocké sur un disque, on utilise fréquemment des valeurs de t de 50 à 2000, qui dépendent du rapport entre la taille d'une clé et la taille d'une page.

La mémoire d'un système informatique:

-) La **mémoire primaire** (ou **mémoire principale**) constituée habituellement de puces en silicone. Le **temps d'acces** aux données est négligable.

-) Un système possède un général de la **mémoire secondaire** sous forme de disques magnétiques (d'un à centaine gigabit).

-) Le **temps d'acces** aux données peut être important (environ 20 millisecondes).

-) La quantité de mémoire disponible sur ces espaces secondaires dépasse souvent de plusieurs ordres de grandeur la capacité de la mémoire principale.

Lecteur de disque classique:

-) La **tête de lecture/écriture** est capable de lire ou écrire les données magnétiquement sur la surface en rotation.

-) Le **bras de lecture/écriture** peut positionner la tête à différentes distances du centre.

-) Lorsque la tête est stationnaire, la surface qui passe dessous est appellée une **piste**.

-) L'information contenue dans chaque piste est souvent divisée en un nombre fixe de **pages** (ou **secteurs**). Cette page constitué généralement l'unité de base pour le stockage des informations - c'est-à-dire que les lectures et écritures se font le plus souvent sur des pages entières.

La contrepartie du coût réduit des techniques de stockage sur support magnétique est donc le temps relativement long nécessaire pour accéder aux données.

Les unités de stockages entièrement électroniques, de même que les puces en silicone, ont de temps d'accès beaucoup plus réduits que celles qui possèdent des parties mécaniques, comme les lecteurs de disque.

Souvent, il est plus long pour l'ordinateur d'accéder à une page d'information et de la lire sur un disque, que de l'examiner ensuite.

Nous étudierons séparement les deux composantes principales du temps d'accès:

-) le nombre d'acces au disque

-) le temps CPU

Nous allons utiliser le nombre de pages lues ou écrites pour mesurer la le temps d'accès au disque.

8. Opérations fondamentales sur les B-arbres

Nous étudierons les opérations $RECHERCHER$, $CREER$ et $INSERER$.

Pour ces procédures, nous adapterons deux conventions:

-) La racine du B-arbre se trouve toujours en mémoire principale, de manière qu'une opération $LIRE-DISQUE$ ne soit jamais nécessaire pour la racine.

-) Tout nœud passé en paramètre devra avoir subi auparavant une opération $LIRE-DISQUE$.

Les procédures implémentent des algorithmes en "une passe" qui s'exécutent en descendant depuis la racine de l'arbre, sans avoir à revenir en arrière.