Chapitre 4: Structures d'index

Christophe Morvan

Université Paris-est, Marne-la-Vallée

3 novembre 2015

Plan

Progression

Les index

Observation

Les index constituent un outil indispensable pour naviguer dans des structures linéaires de taille considérable.
On a vu qu'il était possible d'utiliser des table d'associations pour constituer des index.
Il est également possible de construire des structures spécialisées pour l'indexation : les structures d'index

Les index

Observation

Les index constituent un outil indispensable pour naviguer dans des structures linéaires de taille considérable.
On a vu qu'il était possible d'utiliser des table d'associations pour constituer des index.
Il est également possible de construire des structures spécialisées pour l'indexation : les structures d'index

Strcture linéaire non indexée

- Recherche séquentielle (cas général – linéaire)
- Recherche dichotomique (structure triée – logarithmique)

Les index

Observation

Les index constituent un outil indispensable pour naviguer dans des structures linéaires de taille considérable.
On a vu qu'il était possible d'utiliser des table d'associations pour constituer des index.
Il est également possible de construire des structures spécialisées pour l'indexation : les structures d'index

Strcture linéaire non indexée

- Recherche séquentielle (cas général – linéaire)
- Recherche dichotomique (structure triée – logarithmique)

Strcture linéaire indexée

- Recherche dans l'index, puis accès direct
- Mise à jour plus coûteuse

Exemple fondamental

Base de donnée ouvrages

No	Titre	Auteur	Année
0	Avis de tempête	Butcher	2000
1	Le Chash	Vance	1968
2	Dune	Herbert	1965
3	L'empire ultime	Sanderson	2010
4	Une forme de guerre	Banks	1986
5	Le monde du fleuve	Farmer	1971
6	Le nom du vent	Rothfuss	2007
7	Le seigneur des anneaux	Tolkien	1954
8	La volonté du dragon	Davoust	2010

Exemple fondamental

Base de donnée ouvrages

No	Titre	Auteur	Année
0	Avis de tempête	Butcher	2000
1	Le Chash	Vance	1968
2	Dune	Herbert	1965
3	L'empire ultime	Sanderson	2010
4	Une forme de guerre	Banks	1986
5	Le monde du fleuve	Farmer	1971
6	Le nom du vent	Rothfuss	2007
7	Le seigneur des anneaux	Tolkien	1954
8	La volonté du dragon	Davoust	2010

Réaliste

- 600000 fiches — 1 Ko par fiche
- 32 Ko par bloc — Près de 20000 blocs

Progression

Définition

Définition

On appelle fichier de données un fichier organisé de façon séquentielle, contenant un des données structurées (sous forme de *fiches*).

Exemple

Le fichier vu précédemment :

No	Titre	Auteur	Année
0	Avis de tempête	Butcher	2000
1	Le Chash	Vance	1968
	...		
6	Le nom du vent	Rothfuss	2007
7	Le seigneur des anneaux	Tolkien	1954
8	La volonté du dragon	Davoust	2010

Index

Clé de recherche

Une clé de recherche est une liste d'un ou plusieurs attributs sur lesquels portent des critères
Type de critères : Clé (valeur exacte), intervalle, préfixe.

Exemple

- Clé de recherche : Auteur – Critère : Vance
- Clé de recherche : Année– Critère : Entre 1999 et 2008
- Clé de recherche : Titre – Critère : Commence par « Le »

Index

Clé de recherche

Une clé de recherche est une liste d'un ou plusieurs attributs sur lesquels portent des critères
Type de critères : Clé (valeur exacte), intervalle, préfixe.

Exemple

- Clé de recherche : Auteur – Critère : Vance
- Clé de recherche : Année– Critère : Entre 1999 et 2008
- Clé de recherche : Titre – Critère : Commence par « Le »

Définition

On appelle fichier d'index, pour un fichier de données et une clé de recherche, un fichier auxiliaire qui permet de naviguer efficacement dans le fichier de données.

Index non-dense

Hypothèse

On suppose que le fichier de données est trié suivant une clé principale qui sert de support pour l'index.

Index non-dense

Hypothèse

On suppose que le fichier de données est trié suivant une clé principale qui sert de support pour l'index.

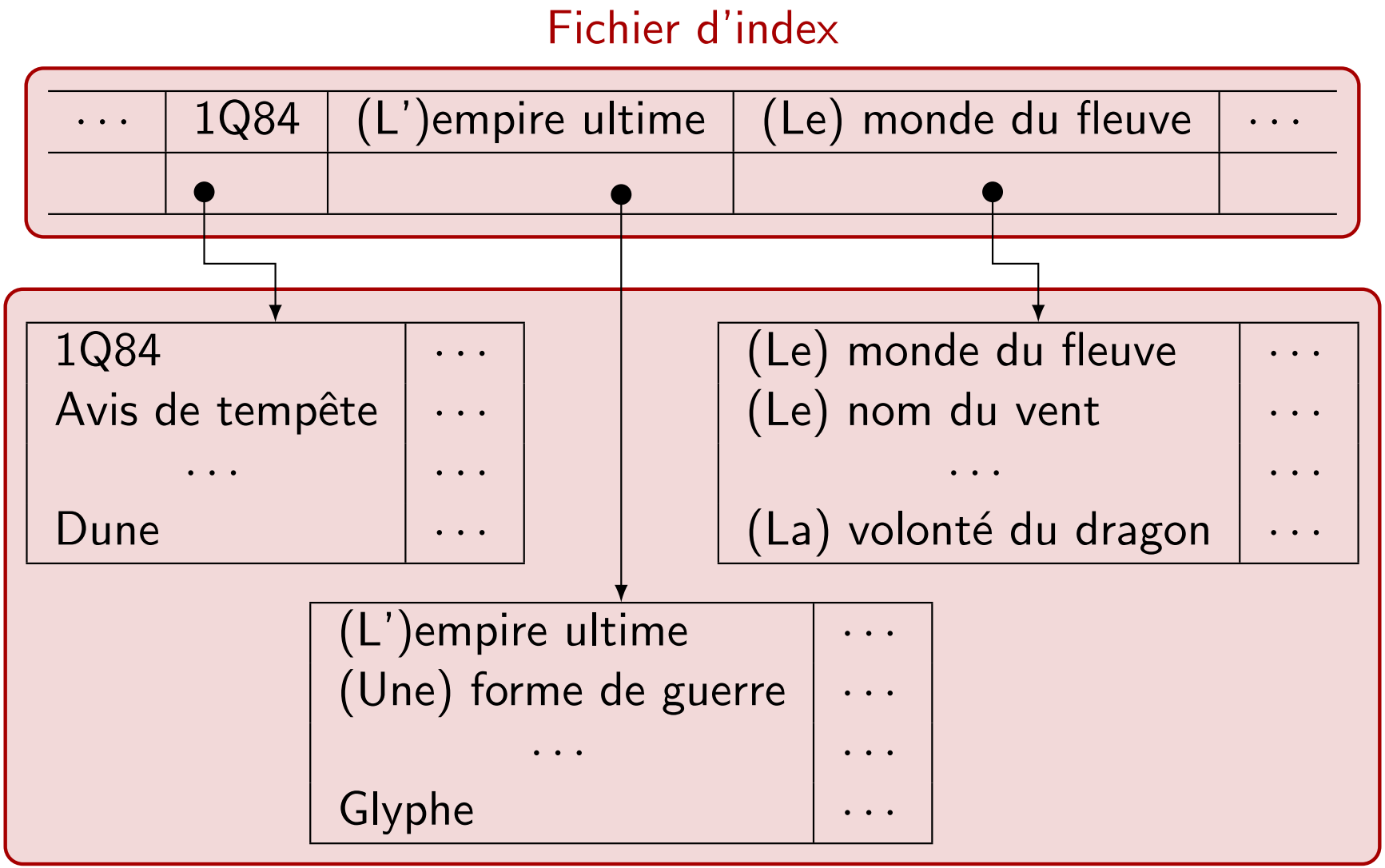
L'index *non-dense*

Un fichier – Contenant des enregistrements : (Valeur, Adresse)

Le fichier de données et le fichier d'index sont triés sur Valeur

Seul le premier enregistrement d'un bloc est stocké dans l'index

Exemple


Fichier d'index
· · ·
1Q84
(L')empire ultime
(Le) monde du fleuve
· · ·
1Q84
Avis de tempête
· · ·
Dune
(Le) monde du fleuve
(Le) nom du vent
· · ·
(La) volonté du dragon
(L')empire ultime
(Une) forme de guerre
· · ·
Glyphe
Fichier de données

Opérations

Recherches

- **Par clé** : Recherche dichotomique sur l'index.
- **Par intervalle** :
  - Recherche dichotomique de la borne inférieure dans l'index.
  - Parcours séquentiel du fichier.
- **Par préfixe** : cas particulier de recherche par intervalle.

Opérations

Recherches

- **Par clé** : Recherche dichotomique sur l'index.
- **Par intervalle** :
  - Recherche dichotomique de la borne inférieure dans l'index.
  - Parcours séquentiel du fichier.
- **Par préfixe** : cas particulier de recherche par intervalle.

Programme exemple

```
static int rechDicho(int val, int [] tab){
    int min=0, max=tab.length-1, mid=0;
    while(min<=max){
        mid=(max+min)/2;
        if (val==tab[mid])return mid;
        else if (val>tab[mid])min=mid+1;
        else max = mid-1;}
    return -1;}
```

Données concrètes

Données

- 600000 fiches de 1 Ko (soit 600 Mo)
- On a 600000/32 blocs : 18750
- En comptant 248 octets pour un titre et 8 octets pour une adresse l'index fait environs 5 Mo
- Permet de gagner environs 7 étapes sur la recherche dichotomique

Données concrètes

Données

- 600000 fiches de 1 Ko (soit 600 Mo)
- On a 600000/32 blocs : 18750
- En comptant 248 octets pour un titre et 8 octets pour une adresse l'index fait environs 5 Mo
- Permet de gagner environs 7 étapes sur la recherche dichotomique

Difficulté

Maintenir trié le fichier de données et l'index.

Index dense

Hypothèse

On ne suppose pas que le fichier de données est trié suivant la clé qui sert de support pour l'index.

Index dense

Hypothèse

On ne suppose pas que le fichier de données est trié suivant la clé qui sert de support pour l'index.

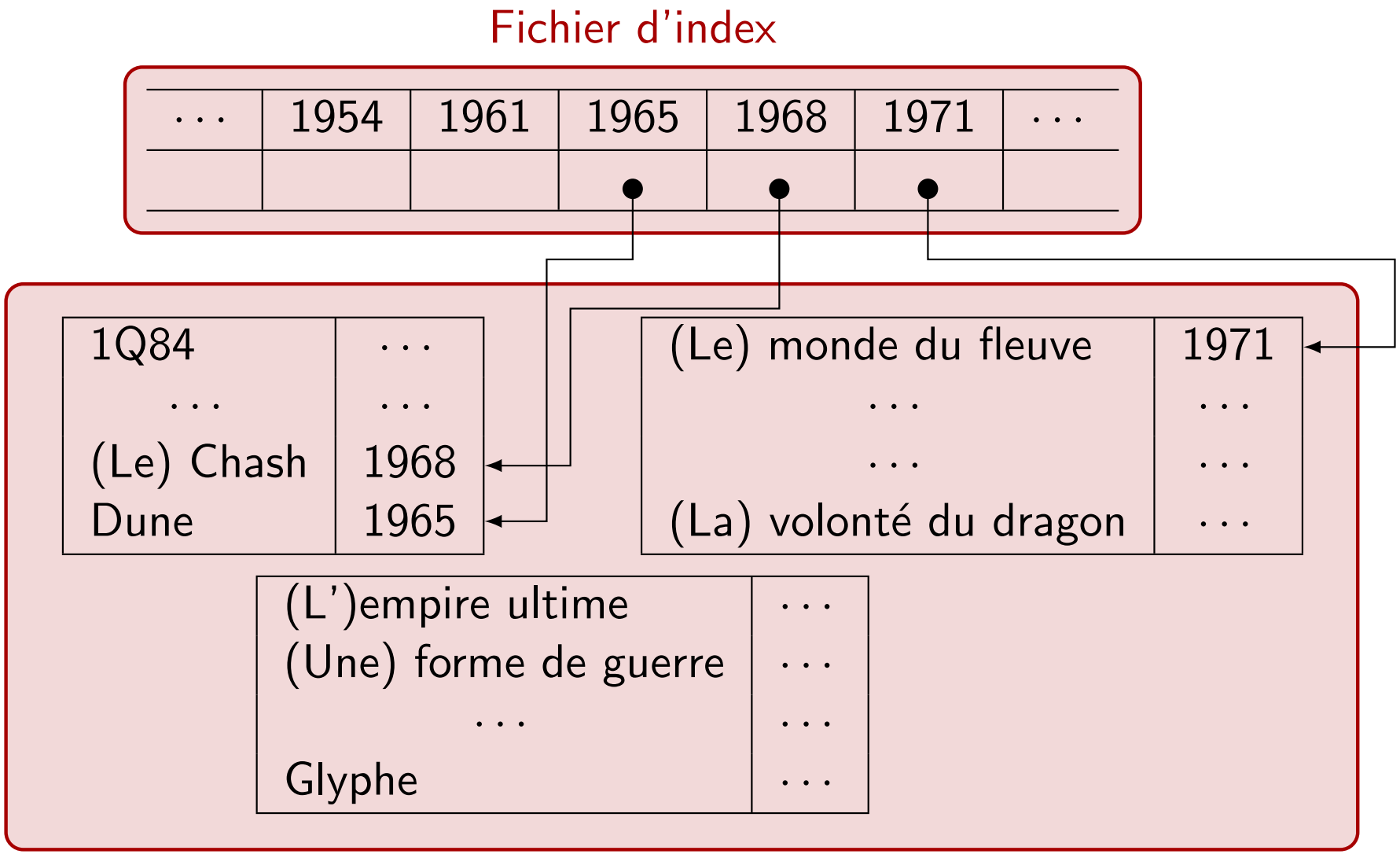
L'index *dense*

Un fichier – Contenant des enregistrements : (Valeur, Adresse)

Le fichier d'index est trié sur Valeur

Tous les enregistrements sont stockés dans l'index

Exemple


Fichier d'index
... 1954 1961 1965 1968 1971 ...
1Q84 ...
... ...
(Le) Chash 1968
Dune 1965
(Le) monde du fleuve 1971
... ...
... ...
(La) volonté du dragon ...
(L')empire ultime ...
(Une) forme de guerre ...
... ...
Glyphe ...
Fichier de données

Opérations

Recherches

- **Par clé** : Recherche dichotomique sur l'index (comme non-dense).
- **Par intervalle** :
  - Recherche dichotomique de la borne inférieure dans l'index.
  - Parcours séquentiel du fichier d'index.
  - Pour chaque valeur accès au fichier de données.

Coût plus élevé.

Opérations

Recherches

- **Par clé** : Recherche dichotomique sur l'index (comme non-dense).
- **Par intervalle** :
  - Recherche dichotomique de la borne inférieure dans l'index.
  - Parcours séquentiel du fichier d'index.
  - Pour chaque valeur accès au fichier de données.

Coût plus élevé.

Programme exemple

```
static int rechDichoMin(int val, int [] tab){
    /* Trouver l'indice de la premiere des valeurs >= a val*/
    int min=0, max=tab.length-1, mid=0;
    while(min<=max){  mid=(max+min)/2;
        if (val==tab[mid])return mid;
        else if (val>tab[mid])min=mid+1;
        else max = mid-1;}
    return min;}
```

Données concrètes

Données

- 600000 fiches de 1 Ko (soit 600 Mo).
- En comptant 4 octets pour une année et 8 octets pour une adresse l'index fait environs $7,2$ Mo.
- La taille de l'index est environs 100 fois moindre que le fichier (comme pour l'index non-dense... si les données individuelles sont moins importantes le facteur de réduction est moindre).

Données concrètes

Données

- 600000 fiches de 1 Ko (soit 600 Mo).
- En comptant 4 octets pour une année et 8 octets pour une adresse l'index fait environs $7,2$ Mo.
- La taille de l'index est environs 100 fois moindre que le fichier (comme pour l'index non-dense... si les données individuelles sont moins importantes le facteur de réduction est moindre).

Bénéfices

- Il n'est pas nécessaire d'avoir un fichier de données trié (permet une insertion aisée dans le fichier de données).
- Permet une recherche dichotomique sur une clé pour laquelle le fichier de données n'est pas trié.
- Permet de disposer de plusieurs clés de recherche en ayant plusieurs fichiers d'index (pour un même fichier de données).

Fichiers d'index

Index multi-niveau

Lorsque la taille de l'index est trop importante on peut indexer l'index.

- L'index est trié on peut donc réaliser un index non-dense
- Si ce n'était pas le cas aucun intérêt (pourquoi ?)
- On peut répéter le processus pour atteindre une page unique.

On appelle le résultat structure séquentielle indexée.

Fichiers d'index

Index multi-niveau

Lorsque la taille de l'index est trop importante on peut indexer l'index.

- L'index est trié on peut donc réaliser un index non-dense
- Si ce n'était pas le cas aucun intérêt (pourquoi ?)
- On peut répéter le processus pour atteindre une page unique.

On appelle le résultat structure séquentielle indexée.

Observations

- On peut avoir un seul index non-dense par fichier (pourquoi ?).
- Il peut y avoir plusieurs index denses.

Progression

Arbres B

Observation

Les structures d'index nécessitent des structures de données permettant de maintenir efficacement triées des volumes important de données.

Arbres B

Observation

Les structures d'index nécessitent des structures de données permettant de maintenir efficacement triées des volumes important de données.

Arbres B

Un arbre B est un arbre équilibré dont les nœuds sont des fragments d'index. Il se réorganise dynamiquement au fil des insertions et des supression.
(Un peu comme l'arbre rouge-noir)

Arbres B

Observation

Les structures d'index nécessitent des structures de données permettant de maintenir efficacement triées des volumes important de données.

Arbres B

Un arbre B est un arbre équilibré dont les nœuds sont des fragments d'index. Il se réorganise dynamiquement au fil des insertions et des supression.
(Un peu comme l'arbre rouge-noir)

Utilisation

- Base de données
- Systèmes de fichiers
  (Btrfs en particulier)

Variante

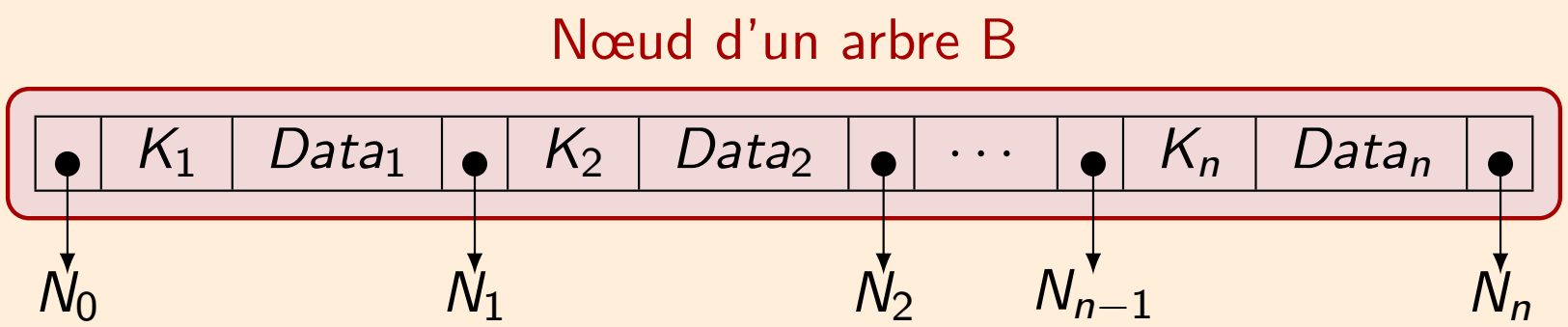
Les arbres B+ :
Dans plusieurs systèmes de fichiers :
NTFS, ReiserFS, JFS, XFS, ...

Nœuds d'un arbre B

Détail

Chaque nœud contient une alternance entre des clés (avec une référence ou le contenu d'un enregistrement) et des pointeurs vers les nœuds suivants. Les clés sont positionnées suivant un ordre, et les pointeurs désignent des nœud dont le sous-arbre contient toutes les clés se plaçant entre les deux clés voisine du pointeur.

Exemple



Le sous-arbre du nœud N_1 contient tous les éléments dont la clé est comprise entre K_1 et K_2.

Construction d'un arbre B

Paramètres

On fixe le nombre maximal d'enregistrement par nœud
On fixe le nombre minimal de clé (en général à la moitié du max)

Insertion

1) On trouve la place de l'élément à insérer.
2) Si le nœud peut accueillir le nouvel élément c'est terminé
3) Si jamais le nœud ne peut accueillir le nouvel élément on le subdivise en deux et on remonte la valeur médiane au niveau du père (pour séparer les deux fils)
4) On répète récursivement le processus en cas de nécessité.

La suppression est un peu plus complexe on l'examinera en TD.

Arbre B+

Inconvénient principal de l'arbre B

Les enregistrement sont déplacés lors de l'insertion/suppression d'éléments
Problème en cas de pointeurs sur ces éléments

Arbre B+

Il est construit simplement sur les clés
Ces dernières sont exclusivement placées dans les feuilles
Dans les feuilles chaque clé est associé à une adresse

Permet également d'avoir plusieurs arbres B+ pour un seul fichier.
(substituer les enregistrements par des adresses dans l'arbre B résout une partie de ce problème)
Dans les arbres B+ chaque nœud est un bloc contenant un nombre fixé de références vers des sous-arbres
Les feuilles sont liées entre elles ce qui permet un parcours séquentiel entre deux bornes (plus complexe pour l'arbre B)

Synthèse

Structures d'index

Index non-denses

Index denses

Arbres *de recherche*

Arbres B

Arbres B+ (Sommairement)