Structures de données avancées :

LH (Hachage linéaire)

Pr ZEGOUR DJAMEL EDDINE
Ecole Supérieure d'Informatique (ESI)
www.zegour.uuuq.com
email: d_zegour@esi.dz

LH : Hachage linéaire

- **LH utilise les fonctions h_0, h_1, h_2, ... telles que (fonctions de division)**

  - h_i : C → {0, 1, ..., 2^i.N - 1}

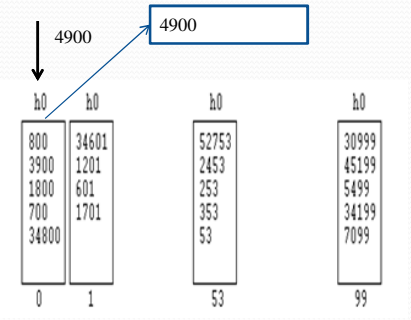
  - h_i (c) = h_{i-1} (c)
  ou bien
  - h_i (c) = h_{i-1} (c) + 2^{i-1} .N

- **Fonctions couramment utilisées :**

  h_i (c) = c Mod 2^i .N

LH : Hachage linéaire

- **Construction de la table avec la méthode classique**
- **L'insertion de 4900 provoque une collision sur la case o.**
- **Pour le chainage séparé, la clé est rangée en débordement**



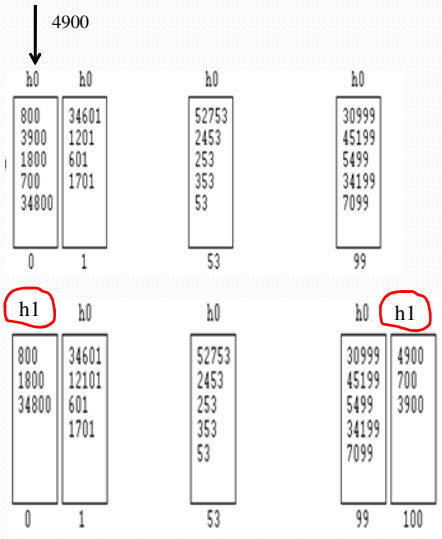
h = c Mod N ; N = 100, capacité b = 5

LH : Hachage linéaire

- **Construction de la table avec la méthode classique (fonction h_o)**
- **L'insertion de 4900 provoque une collision sur la case o.**
- **Le fichier est donc réorganisé selon la nouvelle h.**

  $h(c) = h_1(c)$ si $h_o(c) = o$

  $h(c) = h_o(c)$ sinon.



h0 = c Mod N ; N = 100, capacité b = 5

LH : Hachage linéaire

Idée principale

- **L'application des fonctions de division a résolu la collision**
  SANS création de débordement
  SANS dégradation des performances d'accès.

- **Le fichier est étendu d'une case.**

LH : Hachage linéaire

Stratégie de division

- **A chaque collision il y a**
  - Remplacement de la fonction courante
  - Réorganisation correspondante.

- **Le principe de LH est de diviser les cases dans un ordre prédéfini.**

P = **0, 1, 2, …., N-1**
0, 1, 2, …., 2N-1
….
0, 1, ….., 2^i.N-1

P : Prochaine case à éclater

LH : Hachage linéaire

Stratégie de division

- **Une séquence est dite "trip" ou voyage.**
- **Le i-ème voyage utilise la fonction h_i.**
- **Le premier voyage remplace de gauche à droite h_0 par h_1 en étendant le fichier de 2N cases.**
- **Le second voyage remplace h_1 par h_2 en étendant le fichier avec 4N cases.**
- **Etc…**

0, 1, 2, …., N-1

0, 1, 2, …., 2N-1

….

0, 1, ….., 2^i.N-1

LH : Hachage linéaire

Stratégie de résolution de collision

- **L'adresse d'une collision est typiquement aléatoire (m)**
- **La situation P=m est rare. (P prochaine case à éclater)**
- ***Règle de division***

✓ **Chaque collision sur m conduit à la division de la case P. (Faire P ← P+1 ou P ← 0)**

✓ **Sauf pour le cas où P=m, la clé est rangée en débordement selon une méthode classique de résolution.**

LH : Hachage linéaire

Initialement le fichier est vide
Capacité de la case : b >> 1
Soit mj l'adresse de la j-ième collision)

Évolution du fichier

- **Première collision**

➢ **Quelque soit la valeur de m1, il y a division de la case 0.**
➢ **Sauf pour m1 = 0, la nouvelle clé est rangée en débordement.**

- **Deuxième collision**

➢ **Quelque soit la valeur de m2, il y a division de la case 1.**
➢ **Sauf pour m2= 1, la nouvelle clé est rangée en débordement.**

Ce traitement continue jusqu'a N collisions.

LH : Hachage linéaire

Évolution du fichier

- **Après N divisions**

✓ **ho est totalement remplacée par h1.**

✓ **La situation est donc analogue à la situation initiale sauf que le fichier contient 2 fois plus de cases.**

✓ **Les 2N prochaines divisions utilisent la fonction h_2.**

✓ **P reprend la valeur 0**

LH : Hachage linéaire

Soit j l'indice (le niveau) de la fonction de division courante.
Soit c une clé

Calcul de l'adresse primaire

$m \leftarrow h_j(c)$
Si $m < P$
    $m \leftarrow h_{j+1}(c)$
Fsi

P étant l'adresse de la prochaine case à éclater.

LH : Hachage linéaire

Soit j l'indice (le niveau) de la fonction de division courante.
Soit c une clé

Calcul de l'adresse primaire

$m \leftarrow h_j(c)$
Si $m \geq M$
    $m \leftarrow h_{j-1}(c)$
Fsi

M étant le nombre courant de cases primaires.

LH : Hachage linéaire

Performances

- **b : capacité de case**
- **a : facteur de chargement**
- **s' : recherche avec succès**
- **s" : recherche sans succès**

b	10	20	50
s''	1.19	1.12	1.06
s'	1.07	1.02	1.00
a	0.61	0.59	0.59

LH contrôlé

Principe

- **L'éclatement d'une case est provoqué quand le facteur de chargement devient supérieur à un seuil s donné.**
- **Dans le cas de non éclatement : utilisation chaînage séparé**
- **Quand une donnée est supprimée, la contraction (opération inverse de l'éclatement)sera faite quand le facteur de chargement devient inférieur ou égal à un seuil s' donné.**

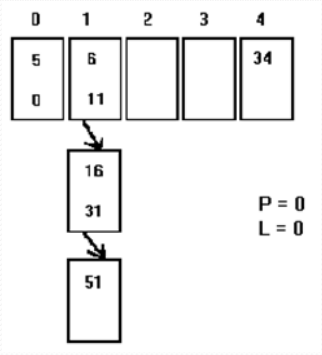
LH contrôlé

Exemple

Paramètres :
- **N = 5**
- **b = b' = 2;**
- **s=0.60, s'=0.50**

$h_i(c) = c \text{ Mod } 2^i.N$
i = 0, au départ

Insertion
11, 0, 5, 6, 16, 31, 34, 51

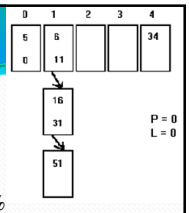


LH contrôlé

Exemple

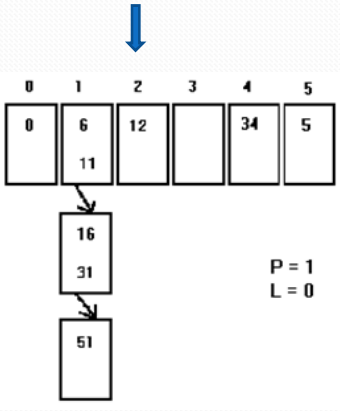
Paramètres :
- **N = 5**
- **b = b' = 2;**
- **s=0.60, s'=0.50**

$h_i(c) = c \text{ Mod } 2^i.N$
i = 0, au départ



Insertion de 12

Facteur de chargement = 64%

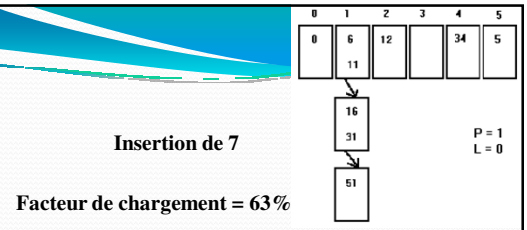


LH contrôlé

Exemple

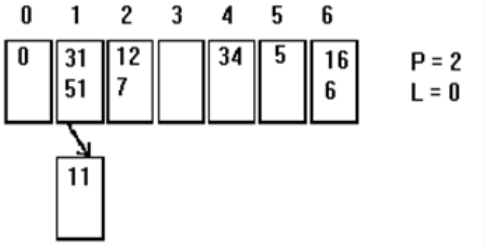
Paramètres :
- **N = 5**
- **b = b' = 2;**
- **s=0.60, s'=0.50**

$h_i(c) = c \text{ Mod } 2^i.N$
i = 0, au départ



Insertion de 7

Facteur de chargement = 63%

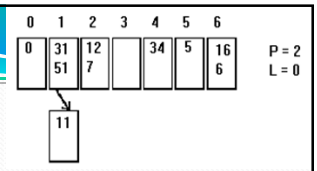


LH contrôlé

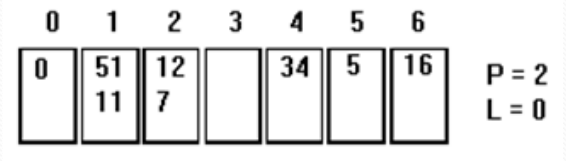
Exemple

Paramètres :
- **N = 5**
- **b = b' = 2;**
- **s=0.60, s'=0.50**

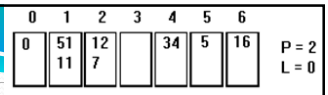
$h_i(c) = c \text{ Mod } 2^i.N$
i = 0, au départ



Suppression de 6, 31



LH contrôlé



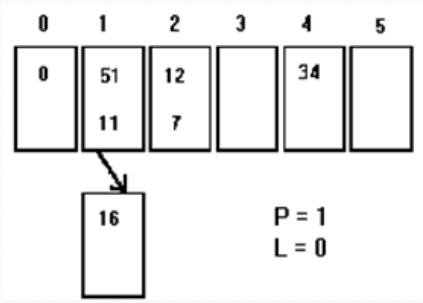
Suppression de 5

Exemple

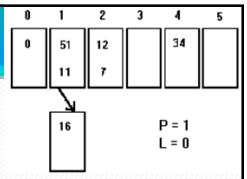
Facteur de chargement = 50%

Paramètres :
- **N = 5**
- **b = b' = 2;**
- **s=0.60, s'=0.50**

$h_i(c) = c \text{ Mod } 2^i.N$
i = 0, au départ



LH contrôlé



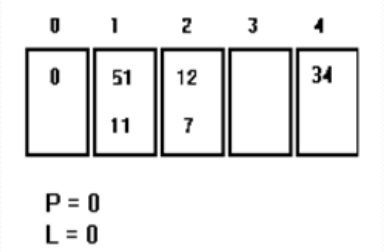
Suppression de 16

Exemple

Facteur de chargement = 50%

Paramètres :
- **N = 5**
- **b = b' = 2;**
- **s=0.60, s'=0.50**

$h_i(c) = c \text{ Mod } 2^i.N$
i = 0, au départ



LH contrôlé

Synthèse :

Fournir

-un facteur de chargement élevé

-avec une légère détérioration des performances d'accès.

***Quelques résultats* : (b=20)**

Avec un chargement de 75%
✓recherche avec succès 1.08
✓recherche sans succès 1.29

***Quelques résultats* : (b=20)**

Avec un chargement de 90%
✓recherche avec succès 1.44
✓recherche sans succès 2.45

LH contrôlé avec expansion partielle

Le Concept

- **Dans LH, les cases éclatées sont moins remplies que les cases non encore éclatées**
- **Uniformiser le chargement est un moyen d'améliorer les performances d'accès.**
- **Avec deux expansions partielles,**
  - **Première expansion: une moitié est rajoutée**
    Deuxième expansion: il double

  - **Première expansion : éclater 2 cases . (Une nouvelle case est créée)**
    Deuxième expansion : éclater 3 cases. (Une nouvelle case est créée)

  - **Première expansion : transférer 1/3 de la case.**
    Deuxième expansion: transférer ¼ de la case.

LH contrôlé avec expansion partielle

Principe

On commence par un fichier de taille 2N.

- **Il est logiquement divisé en N groupes $(j, N+j)$, j=0, 1, ..N-1**
- **On divise d'abord les cases du groupe 1, puis les cases du groupe 2, …**
- **L'éclatement du groupe j déplace à peu près 1/3 des clés des cases j et j+N dans la case nouvellement ajoutée j + 2N.**
- **Quand le dernier groupe est éclaté, le fichier est étendu de 2N à 3N.**

LH contrôlé avec expansion partielle

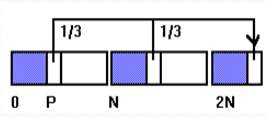
Principe

- **Par la suite, on travaille sur des groupes de 3 pages $(j, N+j, 2N + j)$,j=0, 1, 2…N-1.**
- **L'éclatement du groupe j déplace à peu près 1/4 des clés des cases j, j+N et j+2N dans la case nouvellement ajoutée j + 3N.**
- **Le fichier passe donc de 3N à 4N.**
- **Le processus recommence avec des groupes de 2 cases (j, 2N +j) j= 0, 1, …N-1.**

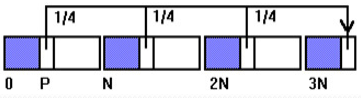
LH contrôlé avec expansion partielle

Illustration de LH avec deux expansions partielles

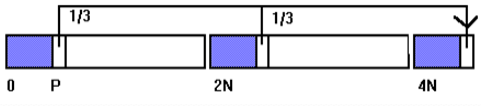
Première expansion



Deuxième expansion



Première expansion



• • •

LH contrôlé avec expansion partielle

Performances

Quelques résultats **: (b=20)**

- **Avec un chargement de 75%**
- ✓ **recherche avec succès 1.03**
- ✓ **recherche sans succès 1.15**

Quelques résultats **: (b=20)**

- **Avec un chargement de 90%**
- ✓ **recherche avec succès 1.18**
- ✓ **recherche sans succès 1.94**

Quelques résultats **: (b=20)**

- **Avec un chargement de 95%**
- ✓ **recherche avec succès 1.45**
- ✓ **recherche sans succès 3.24**

LH contrôlé avec essai linéaire

Concept

- **Dans le schéma de base, les débordements sont rangés dans une zone (fichier) à part (chaînage séparé)**
- **On peut appliquer toute autre méthode du hachage classique**
- **LH contrôlé avec essai linéaire : le plus intéressant (Lectures de plusieurs cases en même temps (Contiguïté))**

Quelques résultats **: (b=20)**
Avec un chargement de 75%

- ✓ **recherche avec succès 1.01**
- ✓ **recherche sans succès 1.13**

LH récursif

Concept

- **Gérer les débordements de la même façon que le fichier principal**
- **Généralement on va jusqu'au 3-ième niveau.**

Quelques résultats **: (b=20)**

- **Avec un chargement de 75%**
- ✓ **recherche avec succès 1.03**
- ✓ **recherche sans succès 1.29**

Quelques résultats **: (b=20)**

- **Avec un chargement de 95%**
- ✓ **recherche avec succès 1.29**
- ✓ **recherche sans succès 3.35**

LH : Conclusion

- **Nombre d'accès disque au voisinage de 1 quelque soit le nombre d'articles insérés.**
- **Fichiers dynamiques non ordonnés**
- **Facteur de chargement contrôlé**