



Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Les tables de hachage

Nour-Eddine Oussous, Éric Wegrzynowski

Licence ST-A, USTL - API2

26 novembre 2008



Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

1 Introduction

2 Proposition de solution

- Représentation par tableau
- Représentation par tableau d'enregistrements

3 Solution : Table de hachage

- Table de hachage
- Fonction de hachage
- Résolution des collisions
- Choix de la fonction de hachage



Introduction

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Dans de nombreuses applications, on fait appel aux *ensembles dynamiques* qui ne supportent que les opérations de *dictionnaire*



Introduction

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Dans de nombreuses applications, on fait appel aux *ensembles dynamiques* qui ne supportent que les opérations de *dictionnaire*
 - Insérer(x , T)



Introduction

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Dans de nombreuses applications, on fait appel aux *ensembles dynamiques* qui ne supportent que les opérations de *dictionnaire*
 - Insérer(x , T)
 - Supprimer(x , T)



Introduction

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Dans de nombreuses applications, on fait appel aux *ensembles dynamiques* qui ne supportent que les opérations de *dictionnaire*
 - Insérer(x , T)
 - Supprimer(x , T)
 - Rechercher(x , T)



Introduction

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Dans de nombreuses applications, on fait appel aux *ensembles dynamiques* qui ne supportent que les opérations de *dictionnaire*
 - Insérer(x, T)
 - Supprimer(x, T)
 - Rechercher(x, T)
- Exemples :
 - *Table des symboles* tenue par un compilateur. Les identificateurs définis dans un programme sont placés dans une table de hachage (avec les info. qui leur sont associées).



Introduction

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Dans de nombreuses applications, on fait appel aux *ensembles dynamiques* qui ne supportent que les opérations de *dictionnaire*
 - Insérer(x, T)
 - Supprimer(x, T)
 - Rechercher(x, T)
- Exemples :
 - *Table des symboles* tenue par un compilateur.
 - *Annuaire téléphonique* qui lie les noms aux numéros de téléphone.



Introduction

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Dans de nombreuses applications, on fait appel aux *ensembles dynamiques* qui ne supportent que les opérations de *dictionnaire*
 - Insérer(x, T)
 - Supprimer(x, T)
 - Rechercher(x, T)
- Exemples :
 - *Table des symboles* tenue par un compilateur.
 - *Annuaire téléphonique* qui lie les noms aux numéros de téléphone.
 - *Table des numéros IP* et des adresses Internet



Introduction

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Dans chacun des exemples précédents, l'information stockée est un *enregistrement* avec une *clé* qui permet de l'identifier de manière unique.



Introduction

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

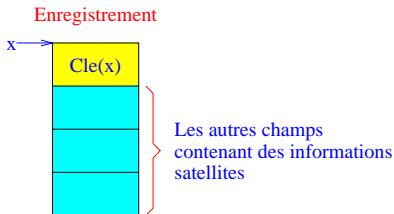
Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Dans chacun des exemples précédents, l'information stockée est un *enregistrement* avec une *clé* qui permet de l'identifier de manière unique.





Introduction

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

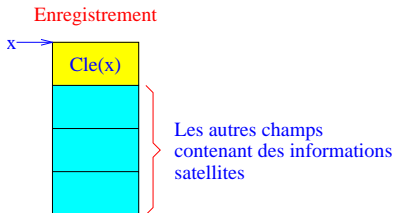
Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Dans chacun des exemples précédents, l'information stockée est un *enregistrement* avec une *clé* qui permet de l'identifier de manière unique.



- On doit définir une structure de données pour organiser ces tables.



Représentation par tableau

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- **Idée** : On suppose que l'ensemble des clés est $K \subset \{0, 1, \dots, m-1\}$ et que les clés sont distinctes. On utilise alors un tableau $T[0..m-1]$:

$$T[k] = \begin{cases} x & \text{si } Cle(x) = k, \\ \text{Null} & \text{sinon} \end{cases}$$

Dans ce cas les opérations sont en $\Theta(1)$.



Représentation par tableau

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- **Idée** : On suppose que l'ensemble des clés est $K \subset \{0, 1, \dots, m-1\}$ et que les clés sont distinctes. On utilise alors un tableau $T[0..m-1]$:

$$T[k] = \begin{cases} x & \text{si } Cle(x) = k, \\ \text{Null} & \text{sinon} \end{cases}$$

Dans ce cas les opérations sont en $\Theta(1)$.

- **Problème** : si l'ensemble des clés ne peut pas servir comme indice du tableau.



Représentation par tableau

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- **Idée** : On suppose que l'ensemble des clés est $K \subset \{0, 1, \dots, m-1\}$ et que les clés sont distinctes. On utilise alors un tableau $T[0..m-1]$:

$$T[k] = \begin{cases} x & \text{si } Cle(x) = k, \\ \text{Null} & \text{sinon} \end{cases}$$

Dans ce cas les opérations sont en $\Theta(1)$.

- **Problème** : si l'ensemble des clés ne peut pas servir comme indice du tableau.
 - L'intervalle des entiers sur 64-bits $\approx 18 \times 10^{18}$ clés possibles. A chaque clé possible est associée une case même si l'enregistrement correspondant est inexistant.



Représentation par tableau

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- **Idée** : On suppose que l'ensemble des clés est $K \subset \{0, 1, \dots, m-1\}$ et que les clés sont distinctes. On utilise alors un tableau $T[0..m-1]$:

$$T[k] = \begin{cases} x & \text{si } Cle(x) = k, \\ \text{Null} & \text{sinon} \end{cases}$$

Dans ce cas les opérations sont en $\Theta(1)$.

- **Problème** : si l'ensemble des clés ne peut pas servir comme indice du tableau.
 - L'intervalle des entiers sur 64-bits $\approx 18 \times 10^{18}$ clés possibles.
 - Chaînes de caractères (encore plus grand !). **L'ensemble des chaînes de caractères sur un alphabet fini est potentiellement infini.**



Représentation par tableau

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- **Idée** : On suppose que l'ensemble des clés est $K \subset \{0, 1, \dots, m-1\}$ et que les clés sont distinctes. On utilise alors un tableau $T[0..m-1]$:

$$T[k] = \begin{cases} x & \text{si } Cle(x) = k, \\ \text{Null} & \text{sinon} \end{cases}$$

Dans ce cas les opérations sont en $\Theta(1)$.

- **Problème** : si l'ensemble des clés ne peut pas servir comme indice du tableau.
 - L'intervalle des entiers sur 64-bits $\approx 18 \times 10^{18}$ clés possibles.
 - Chaînes de caractères (encore plus grand !).



Représentation par tableau

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Conclusion

Avec cette représentation, on réserve beaucoup plus de place qu'il n'en faut !

Pour stocker, par exemple, 10 *enregistrements* de clés entières sur 64 bits, on réserve 18×10^{18} *emplacements* !



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

On stocke les enregistrements dans un tableau et celui-ci est
indiqué arbitrairement.



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

On stocke les enregistrements dans un tableau et celui-ci est
indiqué arbitrairement.

Deux solutions :



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

On stocke les enregistrements dans un tableau et celui-ci est indicé arbitrairement.

Deux solutions :

- le tableau *n'est pas ordonné*



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

On stocke les enregistrements dans un tableau et celui-ci est indicé arbitrairement.

Deux solutions :

- le tableau *n'est pas ordonné*
- on maintient le tableau *ordonné* (selon l'un ou l'autre des champs, la clé par exemple)



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Non ordonné

- La *Recherche* : par parcours séquentiel $\Rightarrow \Theta(n)$.



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Non ordonné

- La *Recherche* : par parcours séquentiel $\Rightarrow \Theta(n)$.

Dans un tableau de 10000 éléments, elle nécessite jusqu'à 10000 comparaisons 5000 en moyenne.



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Non ordonné

- La *Recherche* : par parcours séquentiel $\Rightarrow \Theta(n)$.
- L'*insertion* peut se faire sur le premier emplacement libre et ne demande aucun déplacement $\Rightarrow \Theta(1)$.



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Non ordonné

- La *Recherche* : par parcours séquentiel $\Rightarrow \Theta(n)$.
- L'*insertion* peut se faire sur le premier emplacement libre et ne demande aucun déplacement $\Rightarrow \Theta(1)$.
- La *suppression* peut se faire par déplacement d'une seule entrée $\Rightarrow \Theta(1)$.



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Ordonné selon une clé

- La *recherche* d'un enregistrement, à partir de sa clé, se fait par dichotomie : $\Theta(\log n)$.



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Ordonné selon une clé

- La *recherche* d'un enregistrement, à partir de sa clé, se fait par dichotomie : $\Theta(\log n)$.

Sur un tableau de 10000 entrées, elle nécessite 14 comparaisons.



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Ordonné selon une clé

- La *recherche* d'un enregistrement, à partir de sa clé, se fait par dichotomie : $\Theta(\log n)$.
- L'*insertion* nécessite de déplacer jusqu'à n entrées.



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Ordonné selon une clé

- La *recherche* d'un enregistrement, à partir de sa clé, se fait par dichotomie : $\Theta(\log n)$.
- L'*insertion* nécessite de déplacer jusqu'à n entrées.
Si l'on veut insérer un élément au début d'un tableau à n éléments, on doit déplacer les n éléments.



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Ordonné selon une clé

- La *recherche* d'un enregistrement, à partir de sa clé, se fait par dichotomie : $\Theta(\log n)$.
- L'*insertion* nécessite de déplacer jusqu'à n entrées.
- Idem pour la *suppression*.



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Ordonné selon une clé

- La *recherche* d'un enregistrement, à partir de sa clé, se fait par dichotomie : $\Theta(\log n)$.
- L'*insertion* nécessite de déplacer jusqu'à n entrées.
- Idem pour la *suppression*.

Si l'on veut supprimer le premier élément d'un tableau de n éléments, on doit déplacer les $n - 1$ éléments qui restent vers la gauche.



Tableau d'enregistrements

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Ordonné selon une clé

- La *recherche* d'un enregistrement, à partir de sa clé, se fait par dichotomie : $\Theta(\log n)$.
- L'*insertion* nécessite de déplacer jusqu'à n entrées.
- Idem pour la *suppression*.

Remarque

La recherche d'un enregistrement, à partir d'un champs *différent de la clé*, reste identique à ce qu'elle était dans le cas d'un tableau non classé et est donc en $\Theta(n)$.



Conclusion

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Conclusion

Aucune des deux solutions précédentes n'est satisfaisante !



Table de hachage

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Définition

Une *table de hachage* est un tableau dont la taille est liée au nombre d'entrées que l'on veut mémoriser.

Nous la supposons indicée de 0 à $m - 1$. (m désigne donc sa taille).



Table de hachage

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Définition

Une *table de hachage* est un tableau dont la taille est liée au nombre d'entrées que l'on veut mémoriser.

Nous la supposons indicée de 0 à $m - 1$. (m désigne donc sa taille).

- On utilise une *table de hachage* pour stocker l'information.



Table de hachage

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Définition

Une *table de hachage* est un tableau dont la taille est liée au nombre d'entrées que l'on veut mémoriser.

Nous la supposons indicée de 0 à $m - 1$. (m désigne donc sa taille).

- On utilise une *table de hachage* pour stocker l'information.
- L'information sur laquelle se fera la recherche sera appelée *clé*.



Table de hachage

Les tables de hachage

Nour-Eddine Oussous, Éric Węgrzynowski

Plan

Introduction

Proposition de solution

Solution :
Table de hachage

Définition

Une *table de hachage* est un tableau dont la taille est liée au nombre d'entrées que l'on veut mémoriser.

Nous la supposons indicée de 0 à $m - 1$. (m désigne donc sa taille).

- On utilise une *table de hachage* pour stocker l'information.
- L'information sur laquelle se fera la recherche sera appelée *clé*.
- Le *principe du hachage* consiste à utiliser *la valeur* de la clé pour déterminer à quel *indice* sera rangé l'enregistrement.



Fonction de hachage

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Définition

Soit U l'univers de toutes les clés. La *fonction de hachage* est une fonction qui envoie U sur l'ensemble $[0, m - 1]$.



Fonction de hachage

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

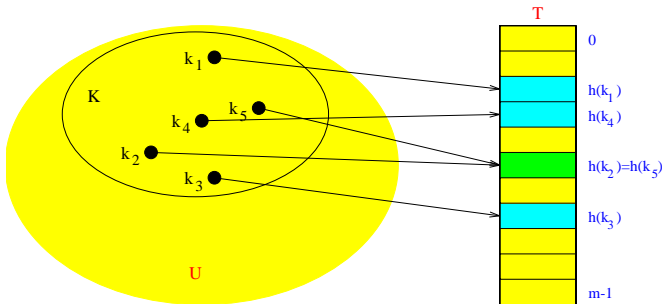
Introduction

Proposition de
solution

Solution :
Table de
hachage

Définition

Soit U l'univers de toutes les clés. La *fonction de hachage* est une fonction qui envoie U sur l'ensemble $[0, m - 1]$.





Fonction de hachage

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

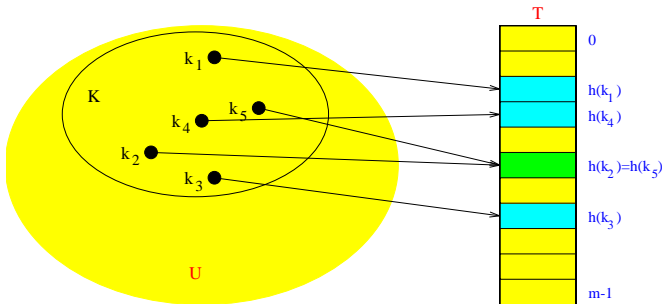
Introduction

Proposition de
solution

Solution :
Table de
hachage

Définition

Soit U l'univers de toutes les clés. La *fonction de hachage* est une fonction qui envoie U sur l'ensemble $[0, m - 1]$.



Quand on veut insérer un enregistrement et que l'image de sa clé est occupée, on dit qu'il y a une *collision*.



Fonction de hachage

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- L'inconvénient de cette solution est qu'une fonction de hachage peut envoyer (*hacher*) plusieurs clés sur le même indice de case et provoquer des *collisions*.



Fonction de hachage

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- L'inconvénient de cette solution est qu'une fonction de hachage peut envoyer (*hacher*) plusieurs clés sur le même indice de case et provoquer des *collisions*.

Exemple

Si les clés sont des valeurs numériques, et que la fonction de hachage est $h(k) = k \bmod m$, alors pour $m = 10000$, on a :

$$h(4325678) = h(25678) = 5678$$



Fonction de hachage

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- L'inconvénient de cette solution est qu'une fonction de hachage peut envoyer (*hacher*) plusieurs clés sur le même indice de case et provoquer des *collisions*.

Exemple

Si les clés sont des valeurs numériques, et que la fonction de hachage est $h(k) = k \bmod m$, alors pour $m = 10000$, on a :

$$h(4325678) = h(25678) = 5678$$

- Il y a différents types de tables de hachage qui se distinguent par la façon de résoudre les collisions.



Résolution des collisions

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Deux méthodes :



Résolution des collisions

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Deux méthodes :

- **Hachage ouvert** : on utilise une table de listes et non une table d'éléments



Résolution des collisions

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Deux méthodes :

- **Hachage ouvert** : on utilise une table de listes et non une table d'éléments
- **Hachage fermé** : on utilise une table d'éléments et on résout les collisions



Hachage ouvert

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

■ Débordement chaîné :



Hachage ouvert

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

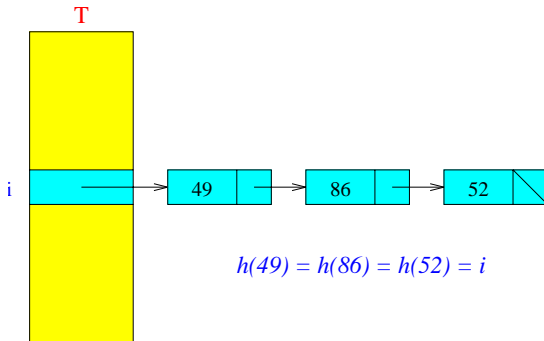
Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

■ Débordement chaîné :





Hachage ouvert

Les tables de hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

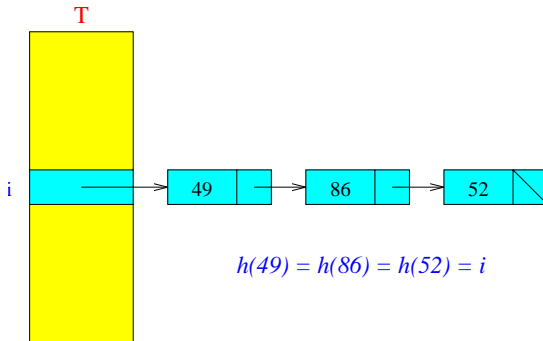
Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

■ Débordement chaîné :



- On construit une *table de listes* et non une table d'éléments.



Les opérations

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Pour cette solution, si h est la fonction de hachage, les opérations de dictionnaire sont faciles à implémenter :

- `Inserer(x, T)` insère x en tête de la liste `T(h(cle(x)))`



Les opérations

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Pour cette solution, si h est la fonction de hachage, les opérations de dictionnaire sont faciles à implémenter :

- **Inserer(x , T)** insère x en tête de la liste **$T(h(cle(x)))$**
Temps de calcul en $\mathcal{O}(1)$ dans le pire des cas.



Les opérations

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Pour cette solution, si h est la fonction de hachage, les opérations de dictionnaire sont faciles à implémenter :

- **Inserer(x , T)** insère x en tête de la liste $T(h(cle(x)))$
Temps de calcul en $\mathcal{O}(1)$ dans le pire des cas.
- **Rechercher(k , T)** recherche un élément de clé k dans la liste $T(h(k))$



Les opérations

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Pour cette solution, si h est la fonction de hachage, les opérations de dictionnaire sont faciles à implémenter :

- **Inserer(x , T)** insère x en tête de la liste $T(h(cle(x)))$
Temps de calcul en $\mathcal{O}(1)$ dans le pire des cas.
- **Rechercher(k , T)** recherche un élément de clé k dans la liste $T(h(k))$
Temps de calcul proportionnel à la longueur de la liste



Les opérations

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Pour cette solution, si h est la fonction de hachage, les opérations de dictionnaire sont faciles à implémenter :

- **Inserer(x , T)** insère x en tête de la liste $T(h(cle(x)))$
Temps de calcul en $\mathcal{O}(1)$ dans le pire des cas.
- **Rechercher(k , T)** recherche un élément de clé k dans la liste $T(h(k))$
Temps de calcul proportionnel à la longueur de la liste
- **Supprimer(x , T)** supprime x de la liste $T(h(cle(x)))$
Temps de calcul en $\mathcal{O}(n)$.



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Exemple

$m = 11$. Clés : $\{15, 33, 39, 40, 63, 116, 118, 123, 11004\}$.

$$h(k) = k \bmod m$$



Exemple

Les tables de hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	



Exemple

Les tables de hachage

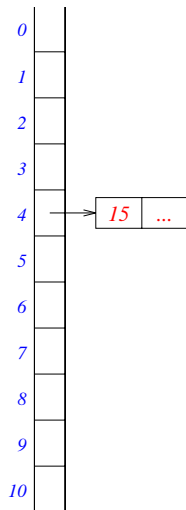
Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage





Exemple

Les tables de hachage

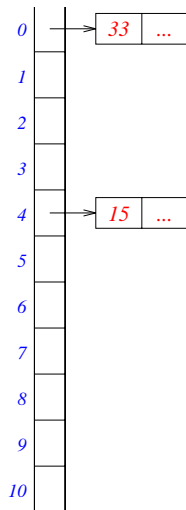
Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage





Exemple

Les tables de hachage

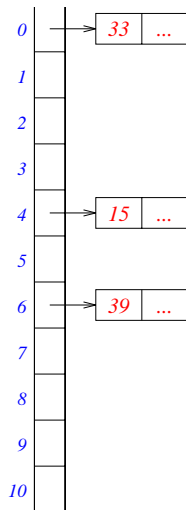
Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage





Exemple

Les tables de hachage

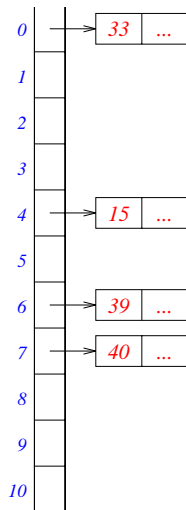
Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage





Exemple

Les tables de hachage

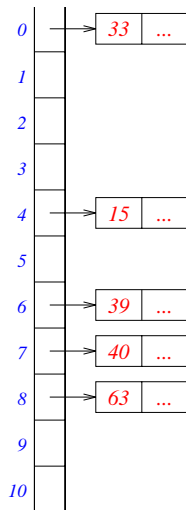
Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage





Exemple

Les tables de hachage

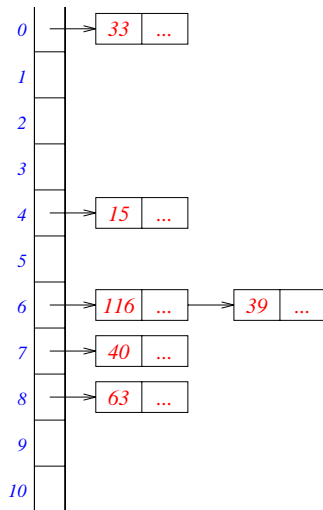
Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage





Exemple

Les tables de hachage

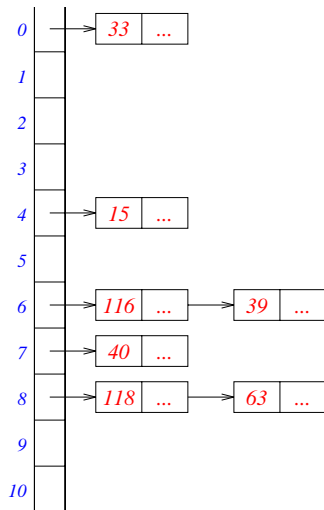
Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage





Exemple

Les tables de hachage

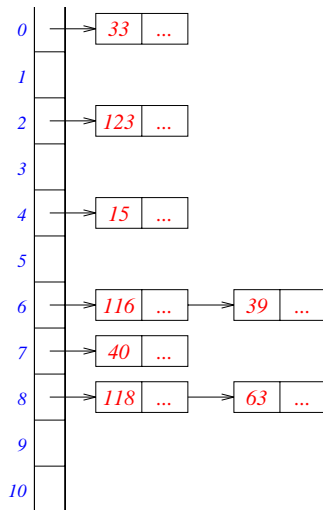
Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage





Exemple

Les tables de hachage

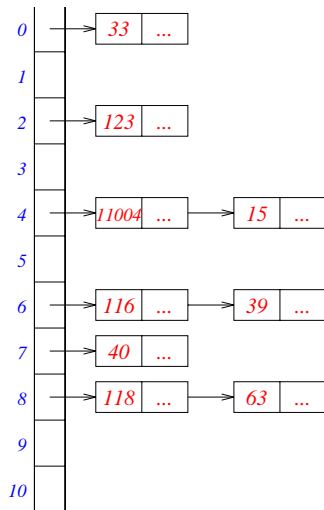
Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage





Analyse de cette solution

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

On prend comme hypothèse que le *hachage est simple et uniforme* :



Analyse de cette solution

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

On prend comme hypothèse que le *hachage est simple et uniforme* :

hachage simple et uniforme

Toute clé $k \in K$ a la même chance d'être hachée dans l'une quelconque des cases indépendamment de l'endroit où les autres clés sont tombées



Probabilité d'existence de collision

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Avec les mêmes notations et h la fonction de hachage uniforme
ie

$$\forall i \in [0, m - 1] \quad \Pr(h(k) = i) = \frac{1}{m} \quad \Pr \equiv \text{Probabilité}$$



Probabilité d'existence de collision

Les tables de hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Avec les mêmes notations et h la fonction de hachage uniforme ie

$$\forall i \in [0, m-1] \quad \Pr(h(k) = i) = \frac{1}{m} \quad \Pr \equiv \text{Probabilité}$$

Dans ce cas h prend n valeurs distinctes pour les n clés. En supposant que les clés ont été choisies au hasard

Pas de collision

La probabilité $P(n, m)$ pour qu'il n'y ait pas de collision est

$$P(n, m) = \frac{m}{m} \times \frac{m-1}{m} \times \dots \times \frac{m-n+1}{m} = \prod_{i=0}^{n-1} \left(1 - \frac{i}{m}\right)$$



Probabilité d'existence de collision

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Avec les mêmes notations et h la fonction de hachage uniforme
ie

$$\forall i \in [0, m-1] \quad \Pr(h(k) = i) = \frac{1}{m} \quad \Pr \equiv \text{Probabilité}$$

Pas de collision

La probabilité $P(n, m)$ pour qu'il n'y ait pas de collision est

$$P(n, m) = \frac{m}{m} \times \frac{m-1}{m} \times \dots \times \frac{m-n+1}{m} = \prod_{i=0}^{n-1} \left(1 - \frac{i}{m}\right)$$



Probabilité d'existence de collision

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

On sait que pour tout x réel, $1 - x \leq e^{-x}$. D'où une majoration pour $P(n, m)$:

$$P(n, m) \leq \prod_{i=0}^{n-1} e^{-\frac{i}{m}} = e^{-\frac{\sum_{i=0}^{n-1} i}{m}} = e^{-\frac{n(n-1)}{2m}}$$



Probabilité d'existence de collision

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

On sait que pour tout x réel, $1 - x \leq e^{-x}$. D'où une majoration pour $P(n, m)$:

$$P(n, m) \leq \prod_{i=0}^{n-1} e^{-\frac{i}{m}} = e^{-\frac{\sum_{i=0}^{n-1} i}{m}} = e^{-\frac{n(n-1)}{2m}}$$

Et la minoration de la probabilité d'existence d'(au moins) une collision

$$1 - P(n, m) \geq 1 - e^{-\frac{n(n-1)}{2m}}$$



$P(n, m)$ en fonction de n pour $m = 365$

Les tables de
hachage

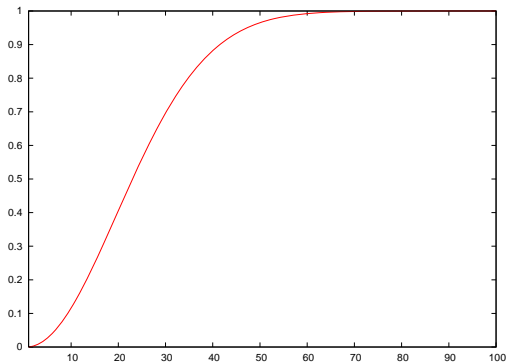
Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage





Analyse de la courbe

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Dès $n = 23$ il y a environ une chance sur deux d'avoir une collision
- Pour $n = 70$ cette probabilité est voisine de 1
- Si on veut maintenir la probabilité d'existence de collision inférieure à un certain seuil β , il est nécessaire d'avoir

$$1 - \beta < e^{-\frac{n(n-1)}{2m}}.$$

n doit donc satisfaire l'inéquation

$$-2m \ln(1 - \beta) > n(n - 1)$$



Analyse de la courbe

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- n est un entier. En posant $K = 2m \ln(1 - \beta)$, n doit être majoré par

$$\frac{1 - \sqrt{1 - 4K}}{2} \sim \sqrt{-2 \ln(1 - \beta)m}$$

- Pour $\beta = \frac{1}{2}$ on a

$$n < 1,17\sqrt{m},$$

- Pour $\beta = \frac{1}{10}$, on obtient

$$n < 0,45\sqrt{m}.$$



Coûts

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Soit n le nombre de clés stockées dans la table et soit m la taille de la table. On définit le *facteur de remplissage* de T par :

$$\alpha = \frac{n}{m} \quad \text{nombre moyen de clés par case}$$



Coûts

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Soit n le nombre de clés stockées dans la table et soit m la taille de la table. On définit le *facteur de remplissage* de T par :

$$\alpha = \frac{n}{m} \quad \text{nombre moyen de clés par case}$$

- Le temps de recherche d'un enregistrement à partir de sa clé est en $\Theta(1 + \alpha)$.
 - 1 calcul de la fonction de hachage et accès.
 - α recherche dans la liste associée à la clé.



Coûts

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Si le nombre de cases de la table est au moins *proportionnel* au nombre d'éléments, on a $n = \mathcal{O}(m)$.
Donc $\alpha = \frac{n}{m} = \frac{\mathcal{O}(m)}{m} = \mathcal{O}(1)$
- Ainsi



Coûts

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Si le nombre de cases de la table est au moins *proportionnel* au nombre d'éléments, on a $n = \mathcal{O}(m)$.
Donc $\alpha = \frac{n}{m} = \frac{\mathcal{O}(m)}{m} = \mathcal{O}(1)$
- Ainsi
 - La recherche est *en moyenne* en $\mathcal{O}(1)$



Coûts

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Si le nombre de cases de la table est au moins *proportionnel* au nombre d'éléments, on a $n = \mathcal{O}(m)$.
Donc $\alpha = \frac{n}{m} = \frac{\mathcal{O}(m)}{m} = \mathcal{O}(1)$
- Ainsi
 - La recherche est *en moyenne* en $\mathcal{O}(1)$
 - L'insertion est *dans le pire des cas* en $\mathcal{O}(1)$



Coûts

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Si le nombre de cases de la table est au moins *proportionnel* au nombre d'éléments, on a $n = \mathcal{O}(m)$.
Donc $\alpha = \frac{n}{m} = \frac{\mathcal{O}(m)}{m} = \mathcal{O}(1)$
- Ainsi
 - La recherche est *en moyenne* en $\mathcal{O}(1)$
 - L'insertion est *dans le pire des cas* en $\mathcal{O}(1)$
 - La suppression est *dans le pire des cas* en $\mathcal{O}(\alpha)$



Coût en place mémoire

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

L'espace mémoire requis par le débordement chaîné est important et constitue l'un de ses inconvénients :



Coût en place mémoire

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

L'espace mémoire requis par le débordement chaîné est important et constitue l'un de ses inconvénients :

- La table contient *m adresses*.



Coût en place mémoire

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

L'espace mémoire requis par le débordement chaîné est important et constitue l'un de ses inconvénients :

- La table contient *m adresses*.
- Les *n* maillons portant les éléments contiennent chacun *1 objet* et *1 adresse*.



Coût en place mémoire

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

L'espace mémoire requis par le débordement chaîné est important et constitue l'un de ses inconvénients :

- La table contient m *adresses*.
- Les n maillons portant les éléments contiennent chacun 1 *objet* et 1 *adresse*.

Soit au total

$$n \text{ objets} + (n + m) \text{ adresses}$$



Coût en place mémoire

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

L'espace mémoire requis par le débordement chaîné est important et constitue l'un de ses inconvénients :

- La table contient m *adresses*.
- Les n maillons portant les éléments contiennent chacun 1 *objet* et 1 *adresse*.

Soit au total

$$n \text{ objets} + (n + m) \text{ adresses}$$

L'autre inconvénient étant le recours à l'allocation dynamique de mémoire (qui peut être lente).



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Tous les éléments sont conservés dans la table de hachage elle-même.



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Tous les éléments sont conservés dans la table de hachage elle-même.

- pour effectuer une insertion, on examine (*sonde*) successivement la table jusqu'à ce qu'on trouve un emplacement libre.



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Tous les éléments sont conservés dans la table de hachage elle-même.

- pour effectuer une insertion, on examine (*sonde*) successivement la table jusqu'à ce qu'on trouve un emplacement libre.
- au lieu de suivre l'ordre $0, 1, \dots, m - 1$, qui demande un temps en $\Theta(m)$, la séquence de positions sondées *dépend de la clé de l'enregistrement à insérer*.



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Tous les éléments sont conservés dans la table de hachage elle-même.

- pour effectuer une insertion, on examine (*sonde*) successivement la table jusqu'à ce qu'on trouve un emplacement libre.
- au lieu de suivre l'ordre $0, 1, \dots, m - 1$, qui demande un temps en $\Theta(m)$, la séquence de positions sondées *dépend de la clé de l'enregistrement à insérer*.
- La fonction de hachage dépend de la clé et du nombre de *sondages* :

$$h : U \times \{0, 1, \dots, m - 1\} \rightarrow \{0, 1, \dots, m - 1\}$$



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Avec le hachage fermé, il faut que la *séquence de sondage* (ou *séquence de parcours*),

$$\langle h(k, 0), h(k, 1), \dots, h(k, m - 1) \rangle$$

soit une permutation de l'ensemble $\{0, 1, \dots, m - 1\}$.

Ainsi, on peut atteindre chacune des cases de la table.



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Avec le hachage fermé, il faut que la *séquence de sondage* (ou *séquence de parcours*),

$$\langle h(k, 0), h(k, 1), \dots, h(k, m - 1) \rangle$$

soit une permutation de l'ensemble $\{0, 1, \dots, m - 1\}$.

- Chaque emplacement de la table finit par être considéré lors du remplissage de la table.



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Avec le hachage fermé, il faut que la *séquence de sondage* (ou *séquence de parcours*),

$$\langle h(k, 0), h(k, 1), \dots, h(k, m - 1) \rangle$$

soit une permutation de l'ensemble $\{0, 1, \dots, m - 1\}$.

- Chaque emplacement de la table finit par être considéré lors du remplissage de la table.
- La table peut être pleine, et les suppressions sont difficiles (mais pas impossibles).



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On ne peut mémoriser plus de m entrées puisque toutes les entrées doivent être stockées directement dans la table



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On ne peut mémoriser plus de m entrées
- Il est nécessaire de pouvoir distinguer les emplacements « *libres* » (ne contenant pas de valeur significative) des autres. Pour ce faire, 2 solutions :



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On ne peut mémoriser plus de m entrées
- Il est nécessaire de pouvoir distinguer les emplacements «*libres*» (ne contenant pas de valeur significative) des autres. Pour ce faire, 2 solutions :
 - Gérer une table de booléens notée **libre**.



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On ne peut mémoriser plus de m entrées
- Il est nécessaire de pouvoir distinguer les emplacements « *libres* » (ne contenant pas de valeur significative) des autres. Pour ce faire, 2 solutions :
 - Gérer une table de booléens notée **libre**.
 - Disposer d'une valeur particulière ne pouvant pas appartenir à U . Cette valeur sera notée symboliquement **Vide**. C'est la solution la plus fréquente.



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On ne peut mémoriser plus de m entrées
- Il est nécessaire de pouvoir distinguer les emplacements «*libres*» (ne contenant pas de valeur significative) des autres. Pour ce faire, 2 solutions :
 - Gérer une table de booléens notée **libre**.
 - Disposer d'une valeur particulière ne pouvant pas appartenir à U . Cette valeur sera notée symboliquement **Vide**. C'est la solution la plus fréquente.
- Par la suite nous utiliserons une fonction
`function Vide(i : Indice) : Boolean ;`
qui indique si une case de la table est vide.



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- En d'autres termes, pour chaque clé possible, on dispose d'un *parcours* de la table.

L'ajout ou la recherche dans la table suit ce parcours.



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- En d'autres termes, pour chaque clé possible, on dispose d'un *parcours* de la table.
L'ajout ou la recherche dans la table suit ce parcours.
- **Un exemple simple** : Le parcours cyclique de un en un est obtenu par la fonction :

$$\begin{aligned}h(k, i) &= (h(k, i - 1) + 1) \bmod m \\&\vdots \\&= (h(k, 0) + i) \bmod m\end{aligned}$$

$h(k, 0)$ définit le point de départ du parcours. **Quand on connaît $h(k, 0)$, on peut calculer les autres positions**



Hachage fermé

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- En d'autres termes, pour chaque clé possible, on dispose d'un *parcours* de la table.
L'ajout ou la recherche dans la table suit ce parcours.
- **Un exemple simple** : Le parcours cyclique de un en un est obtenu par la fonction :

$$\begin{aligned}h(k, i) &= (h(k, i - 1) + 1) \bmod m \\&\vdots \\&= (h(k, 0) + i) \bmod m\end{aligned}$$

$h(k, 0)$ définit le point de départ du parcours.



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Exemple

- On va faire des insertions dans une table de taille $m = 11$ des valeurs

$\{33, 123, 11004, 40, 39, 63, 116, 118, 15\}$

- On prend $h(k, 0) = k \bmod 11$



Exemple

Les tables de hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

0	Vide	...
1	Vide	...
2	Vide	...
3	Vide	...
4	Vide	...
5	Vide	...
6	Vide	...
7	Vide	...
8	Vide	...
9	Vide	...
10	Vide	...



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(33, 0) = 0$$

0	33	...
1	Vide	...
2	Vide	...
3	Vide	...
4	Vide	...
5	Vide	...
6	Vide	...
7	Vide	...
8	Vide	...
9	Vide	...
10	Vide	...



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(123, 0) = 2$$

0	33	...
1	Vide	...
2	123	...
3	Vide	...
4	Vide	...
5	Vide	...
6	Vide	...
7	Vide	...
8	Vide	...
9	Vide	...
10	Vide	...



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(11004, 0) = 4$$

0	33	...
1	Vide	...
2	123	...
3	Vide	...
4	11004	...
5	Vide	...
6	Vide	...
7	Vide	...
8	Vide	...
9	Vide	...
10	Vide	...



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(40, 0) = 7$$

0	33	...
1	Vide	...
2	123	...
3	Vide	...
4	11004	...
5	Vide	...
6	Vide	...
7	40	...
8	Vide	...
9	Vide	...
10	Vide	...



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(39, 0) = 6$$

0	33	...
1	Vide	...
2	123	...
3	Vide	...
4	11004	...
5	Vide	...
6	39	...
7	40	...
8	Vide	...
9	Vide	...
10	Vide	...



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(63, 0) = 8$$

0	33	...
1	Vide	...
2	123	...
3	Vide	...
4	11004	...
5	Vide	...
6	39	...
7	40	...
8	63	...
9	Vide	...
10	Vide	...



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(116, 0) = 6$$

0	33	...
1	Vide	...
2	123	...
3	Vide	...
4	11004	...
5	Vide	...
6	39	...
7	40	...
8	63	...
9	Vide	...
10	Vide	...

collision



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(116, 1) = 7$$

0	33	...
1	Vide	...
2	123	...
3	Vide	...
4	11004	...
5	Vide	...
6	39	...
7	40	...
8	63	...
9	Vide	...
10	Vide	...

collision



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(116, 2) = 8$$

0	33	...
1	Vide	...
2	123	...
3	Vide	...
4	11004	...
5	Vide	...
6	39	...
7	40	...
8	63	...
9	Vide	...
10	Vide	...

collision



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(116, 3) = 9$$

0	33	...
1	Vide	...
2	123	...
3	Vide	...
4	11004	...
5	Vide	...
6	39	...
7	40	...
8	63	...
9	116	...
10	Vide	...

OK



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(118, 0) = 8$$

0	33	...
1	Vide	...
2	123	...
3	Vide	...
4	11004	...
5	Vide	...
6	39	...
7	40	...
8	63	...
9	116	...
10	Vide	...

collision



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(118, 1) = 9$$

0	33	...
1	Vide	...
2	123	...
3	Vide	...
4	11004	...
5	Vide	...
6	39	...
7	40	...
8	63	...
9	116	...
10	Vide	...

collision



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(118, 2) = 10$$

0	33	...
1	Vide	...
2	123	...
3	Vide	...
4	11004	...
5	Vide	...
6	39	...
7	40	...
8	63	...
9	116	...
10	118	...

OK



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(15, 0) = 4$$

0	33	...
1	Vide	...
2	123	...
3	Vide	...
4	11004	...
5	Vide	...
6	39	...
7	40	...
8	63	...
9	116	...
10	118	...

collision



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(15, 1) = 5$$

0	33	...
1	Vide	...
2	123	...
3	Vide	...
4	11004	...
5	15	...
6	39	...
7	40	...
8	63	...
9	116	...
10	118	...

OK



Insertion

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

■ Fonction de hachage :

$$\begin{aligned} h : U \times \{0, 1, \dots, m-1\} &\longrightarrow \{0, 1, \dots, m-1\} \\ (k, i) &\longmapsto h(k, i) \end{aligned}$$



Insertion

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

■ Fonction de hachage :

$$\begin{aligned} h : U \times \{0, 1, \dots, m-1\} &\longrightarrow \{0, 1, \dots, m-1\} \\ (k, i) &\longmapsto h(k, i) \end{aligned}$$

- On examine la table de hachage jusqu'à trouver une case vide.



Insertion

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Fonction de hachage :

$$\begin{aligned} h : U \times \{0, 1, \dots, m-1\} &\longrightarrow \{0, 1, \dots, m-1\} \\ (k, i) &\longmapsto h(k, i) \end{aligned}$$

- On examine la table de hachage jusqu'à trouver une case vide.
- On ne suit pas l'ordre $\{0, 1, \dots, m-1\}$ (en $\Theta(m)$), mais une séquence qui dépend de la clé à insérer.



Insertion

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Fonction de hachage :

$$\begin{aligned} h : U \times \{0, 1, \dots, m-1\} &\longrightarrow \{0, 1, \dots, m-1\} \\ (k, i) &\longmapsto h(k, i) \end{aligned}$$

- On examine la table de hachage jusqu'à trouver une case vide.
- On ne suit pas l'ordre $\{0, 1, \dots, m-1\}$ (en $\Theta(m)$), mais une séquence qui dépend de la clé à insérer.
- La *séquence de parcours*, pour une clé k :

$$\langle h(k, 0), h(k, 1), \dots, h(k, m-1) \rangle$$



Recherche

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Rechercher un enregistrement de clé k consiste à sonder la même séquence que pour l'insertion de l'enregistrement de clé k .



Recherche

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Rechercher un enregistrement de clé k consiste à sonder la même séquence que pour l'insertion de l'enregistrement de clé k .
- La recherche *se termine par un échec* lorsqu'elle rencontre une case qui contient la valeur spéciale **Vide**. Si l'enregistrement de clé k était dans la table, il serait inséré dans cette case et forcément pas plus loin.



Recherche

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Rechercher un enregistrement de clé k consiste à sonder la même séquence que pour l'insertion de l'enregistrement de clé k .
- La recherche *se termine par un échec* lorsqu'elle rencontre une case qui contient la valeur spéciale *Vide*.
- La fonction *retourne l'indice de la case* où se trouve l'enregistrement recherché si celui-ci est présent dans la table.



Suppression

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- La suppression dans la table de hachage à adressage fermé est difficile.



Suppression

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- La suppression dans la table de hachage à adressage fermé est difficile.
- Quand on supprime l'enregistrement qui se trouve à la case *i* de la table, on doit marquer cette case :



Suppression

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- La suppression dans la table de hachage à adressage fermé est difficile.
- Quand on supprime l'enregistrement qui se trouve à la case *i* de la table, on doit marquer cette case :
 - On ne peut pas mettre **Vide** pour la considérer comme toute case vide. **On risque, lors de la recherche, de ne pas trouver les enregistrements qui auraient dû être dans cette case si elle n'était pas occupée et qui ont donc été placés ailleurs.**



Suppression

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- La suppression dans la table de hachage à adressage fermé est difficile.
- Quand on supprime l'enregistrement qui se trouve à la case *i* de la table, on doit marquer cette case :
 - On ne peut pas mettre **Vide** pour la considérer comme toute case vide.
 - On y mettra une valeur spéciale, notée **Oté** par exemple.



Suppression

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- La suppression dans la table de hachage à adressage fermé est difficile.
- Quand on supprime l'enregistrement qui se trouve à la case *i* de la table, on doit marquer cette case :
 - On ne peut pas mettre **Vide** pour la considérer comme toute case vide.
 - On y mettra une valeur spéciale, notée **Oté** par exemple.
- Il faut réécrire la procédure **insérer** pour traiter ce type de case comme si elle était vide.



Suppression

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- La suppression dans la table de hachage à adressage fermé est difficile.
- Quand on supprime l'enregistrement qui se trouve à la case *i* de la table, on doit marquer cette case :
 - On ne peut pas mettre **Vide** pour la considérer comme toute case vide.
 - On y mettra une valeur spéciale, notée **Oté** par exemple.
- Il faut réécrire la procédure **insérer** pour traiter ce type de case comme si elle était vide.
- **Recherche** ne nécessite pas de modification particulière. **Soit on trouve l'élément soit on trouve Vide.**



Choix de la fonction de hachage

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- L'hypothèse « *hachage uniforme simple* » est difficile à garantir, mais il existe des techniques qui fonctionnent bien en pratique.



Choix de la fonction de hachage

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- L'hypothèse « *hachage uniforme simple* » est difficile à garantir, mais il existe des techniques qui fonctionnent bien en pratique.
- Une *bonne fonction de hachage* doit distribuer les clés uniformément dans les cases de la table.



Choix de la fonction de hachage

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- L'hypothèse « *hachage uniforme simple* » est difficile à garantir, mais il existe des techniques qui fonctionnent bien en pratique.
- Une *bonne fonction de hachage* doit distribuer les clés uniformément dans les cases de la table.
- La régularité dans la distribution des clés ne doit pas affecter cette uniformité.



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On suppose que toutes les clés sont des entiers naturels.



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On suppose que toutes les clés sont des entiers naturels.
- A chaque clé k , on associe une case en prenant le reste de la division de k par m :

$$h(k) = k \bmod m$$



Méthode du modulo

Les tables de hachage

Nour-Eddine Oussous, Éric Węgrzynowski

Plan

Introduction

Proposition de solution

Solution :
Table de hachage

- On suppose que toutes les clés sont des entiers naturels.
- A chaque clé k , on associe une case en prenant le reste de la division de k par m :

$$h(k) = k \bmod m$$

- Cette méthode est simple, mais peut donner de mauvaises répartitions si m est mal choisi.



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On suppose que toutes les clés sont des entiers naturels.
- A chaque clé k , on associe une case en prenant le reste de la division de k par m :

$$h(k) = k \bmod m$$

- Cette méthode est simple, mais peut donner de mauvaises répartitions si m est mal choisi.
 - Si m admet un petit diviseur d . La prépondérance des clés congrues modulo d affecte l'uniformité.



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On suppose que toutes les clés sont des entiers naturels.
- A chaque clé k , on associe une case en prenant le reste de la division de k par m :

$$h(k) = k \bmod m$$

- Cette méthode est simple, mais peut donner de mauvaises répartitions si m est mal choisi.
 - Si m admet un petit diviseur d .
 - Si $m = 10^p$ est une puissance de 10. Si les clés sont des nombres en base dix, la fonction de hachage ne dépend que des p chiffres d'ordre inférieur de k .



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On suppose que toutes les clés sont des entiers naturels.
- A chaque clé k , on associe une case en prenant le reste de la division de k par m :

$$h(k) = k \bmod m$$

- Cette méthode est simple, mais peut donner de mauvaises répartitions si m est mal choisi.
 - Si m admet un petit diviseur d .
 - Si $m = 10^p$ est une puissance de 10.
 - Si $m = 2^p$ est une puissance de 2. $h(k)$ ne dépend que des p bits de poids faible de la clé k .
Si $p = 6$ et $k = 1011000111011010_{(2)}$ alors
 $h(k) = 011010_{(2)}$



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On suppose que toutes les clés sont des entiers naturels.
- A chaque clé k , on associe une case en prenant le reste de la division de k par m :

$$h(k) = k \bmod m$$

- Cette méthode est simple, mais peut donner de mauvaises répartitions si m est mal choisi.
 - Si m admet un petit diviseur d .
 - Si $m = 10^p$ est une puissance de 10.
 - Si $m = 2^p$ est une puissance de 2.
Si $p = 6$ et $k = 1011000111\mathbf{011010}_{(2)}$ alors
 $h(k) = 011010_{(2)}$



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On choisit *m premier* assez éloigné des puissances de 2 et des puissances de 10.



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On choisit *m premier* assez éloigné des puissances de 2 et des puissances de 10.
- *Exemple* : On veut une table de hachage pour conserver environ $n = 2000$ chaînes de caractères.



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On choisit *m premier* assez éloigné des puissances de 2 et des puissances de 10.
- **Exemple :** On veut une table de hachage pour conserver environ $n = 2000$ chaînes de caractères.
 - L'examen de 3 éléments en moyenne en cas de recherche infructueuse n'est pas catastrophique



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On choisit m *premier* assez éloigné des puissances de 2 et des puissances de 10.
- **Exemple** : On veut une table de hachage pour conserver environ $n = 2000$ chaînes de caractères.
 - L'examen de 3 éléments en moyenne en cas de recherche infructueuse n'est pas catastrophique
 - On alloue une table de taille $m = 701$



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On choisit *m premier* assez éloigné des puissances de 2 et des puissances de 10.
- **Exemple :** On veut une table de hachage pour conserver environ $n = 2000$ chaînes de caractères.
 - L'examen de 3 éléments en moyenne en cas de recherche infructueuse n'est pas catastrophique
 - On alloue une table de taille $m = 701$
 - 701 est premier
 - 701 est proche de $\frac{2000}{3}$. Autrement dit, $\frac{2000}{701} \approx 3$
 - 701 n'est pas proche d'une puissance de 2
($512 = 2^9 < 701 < 2^{10} = 1024$)



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On choisit *m premier* assez éloigné des puissances de 2 et des puissances de 10.
- **Exemple :** On veut une table de hachage pour conserver environ $n = 2000$ chaînes de caractères.
 - L'examen de 3 éléments en moyenne en cas de recherche infructueuse n'est pas catastrophique
 - On alloue une table de taille $m = 701$
 - 701 est premier
 - 701 est proche de $\frac{2000}{3}$.
 - 701 n'est pas proche d'une puissance de 2 ($512 = 2^9 < 701 < 2^{10} = 1024$)
 - Si l'on traite chaque clé comme un entier, alors
$$h(k) = k \mod 701$$



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Pour des chaînes de caractères, on applique le même principe en prenant pour k le nombre formé en concaténant les caractères.



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Pour des chaînes de caractères, on applique le même principe en prenant pour k le nombre formé en concaténant les caractères.

- Par exemple pour le mot

c	h	a	i	n	e
---	---	---	---	---	---

.

Chaque caractère peut être considéré comme un chiffre dans la base $B = 256$.



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Pour des chaînes de caractères, on applique le même principe en prenant pour k le nombre formé en concaténant les caractères.
- Par exemple pour le mot

c	h	a	i	n	e
---	---	---	---	---	---

.
- Ainsi, dans la base 256 :

$$h(\text{chaîne}) = \left(\begin{array}{l} \text{code}('c') \times 256^5 \\ + \text{code}('h') \times 256^4 \\ + \text{code}('a') \times 256^3 \\ + \text{code}('i') \times 256^2 \\ + \text{code}('n') \times 256^1 \\ + \text{code}('e') \times 256^0 \end{array} \right) \bmod m$$



Méthode du modulo

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Plus généralement,

$$h(C) = \left(\sum_{i=1}^{|C|} \text{code}(C_i) \cdot B^{i-1} \right) \bmod m$$

Qui peut s'implémenter par

```
function hash(const s: STRING) : INDICE ;  
const B= 256 ;  
var R: INDICE ; //implémenté sur au moins 16bits!  
    I: CARDINAL ;  
begin  
    R := 0 ;  
    for I := low(S) to high(S) do  
        R := (R*B + ord(S[I])) mod m ;  
    hash := R ;  
end ; // hash
```

- En fait, on peut aussi prendre pour B une valeur plus petite.



Méthode de multiplication

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On suppose que toutes *les clés sont des nombres entiers*, que $m = 2^r$ et que l'ordinateur travaille avec des mots de w bits.



Méthode de multiplication

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On suppose que toutes *les clés sont des nombres entiers*, que $m = 2^r$ et que l'ordinateur travaille avec des mots de w bits.
- On choisit un nombre $0 < A < 1$ et on définit la fonction de hachage par :

$$h(k) = \lfloor m \cdot (k \cdot A \bmod 1) \rfloor$$

$k \cdot A \bmod 1$ = partie fractionnaire de $k \cdot A$.

Par exemple, $11.03015 \bmod 1 = 0.03015$.



Méthode de multiplication

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- On suppose que toutes *les clés sont des nombres entiers*, que $m = 2^r$ et que l'ordinateur travaille avec des mots de w bits.
- On choisit un nombre $0 < A < 1$ et on définit la fonction de hachage par :

$$h(k) = \lfloor m \cdot (k \cdot A \bmod 1) \rfloor$$

$k \cdot A \bmod 1$ = partie fractionnaire de $k \cdot A$.

Par exemple, $11.03015 \bmod 1 = 0.03015$.

- *Comment implanter h ?*



Méthode de multiplication

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Pour implanter $h(k)$,

- on prend $m = 2^r$. On suppose que *la taille d'un mot machine est w* et que k tient sur 1 mot machine.



Méthode de multiplication

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Pour implanter $h(k)$,

- on prend $m = 2^r$. On suppose que *la taille d'un mot machine est w* et que k tient sur 1 mot machine.
- on calcule $\lfloor A \cdot 2^w \rfloor$: ce sont les w premiers chiffres de la partie fractionnaire de A .
 $0.0110110111 \cdot 2^7 = 0110110.$



Méthode de multiplication

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Pour implanter $h(k)$,

- on prend $m = 2^r$. On suppose que *la taille d'un mot machine est w* et que k tient sur 1 mot machine.
- on calcule $\lfloor A \cdot 2^w \rfloor$: ce sont les w premiers chiffres de la partie fractionnaire de A .
- on calcule $k \cdot \lfloor A \cdot 2^w \rfloor = r_1 2^w + r_0$. Un nombre sur $2w$ bits. Si $k = 1011101$ alors
$$k \cdot \lfloor A \cdot 2^w \rfloor = \underbrace{0100111}_{r_1} \underbrace{0011110}_{r_0}$$



Méthode de multiplication

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Pour implanter $h(k)$,

- on prend $m = 2^r$. On suppose que *la taille d'un mot machine est w* et que k tient sur 1 mot machine.
- on calcule $\lfloor A \cdot 2^w \rfloor$: ce sont les w premiers chiffres de la partie fractionnaire de A .
- on calcule $k \cdot \lfloor A \cdot 2^w \rfloor = r_1 2^w + r_0$.
- on prend les r bits les plus significatifs de r_0 et ce sera $h(k)$.

$$k \cdot \lfloor A \cdot 2^w \rfloor = \underbrace{0100111}_{r_1} \overbrace{001}^r \underbrace{1110}_{r_0}. \text{ Si } r = 3, \text{ on prendra}$$
$$h(k) = 001.$$



Méthode de multiplication

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

Pour implanter $h(k)$,

- on prend $m = 2^r$. On suppose que *la taille d'un mot machine est w* et que k tient sur 1 mot machine.
- on calcule $\lfloor A \cdot 2^w \rfloor$: ce sont les w premiers chiffres de la partie fractionnaire de A .
- on calcule $k \cdot \lfloor A \cdot 2^w \rfloor = r_1 2^w + r_0$.
- on prend les r bits les plus significatifs de r_0 et ce sera $h(k)$.



Méthode de multiplication

Les tables de
hachage

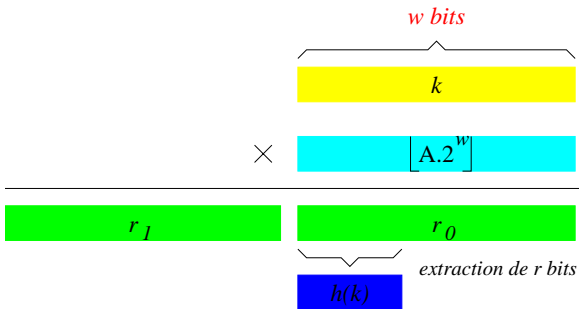
Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage





Méthode de multiplication

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Cette méthode fonctionne mieux pour certaines valeurs de A que pour d'autres. D.E. Knuth a étudié le problème et suggère

$$A \approx \frac{\sqrt{5} - 1}{2} = 0.6180339887 \dots$$



Méthode de multiplication

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Cette méthode fonctionne mieux pour certaines valeurs de A que pour d'autres. D.E. Knuth a étudié le problème et suggère

$$A \approx \frac{\sqrt{5} - 1}{2} = 0.6180339887 \dots$$

- Par exemple, si $k = 123456$ et $m = 10000$, alors

$$\begin{aligned} h(k) &= \lfloor 10000 \cdot (123456 \cdot 0.6180339887 \dots \bmod 1) \rfloor \\ &= \lfloor 10000 \cdot (76300.0041151 \dots \bmod 1) \rfloor \\ &= \lfloor 10000 \cdot 0.0041151 \dots \rfloor \\ &= \lfloor 41.151 \dots \rfloor \\ &= 41 \end{aligned}$$



Méthode de multiplication

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Une autre façon de définir $h(k)$:

$$h(k) = (A \cdot k \bmod 2^w) \text{ dad}(w - r)$$

où A est un *entier impair* tel que $2^{w-1} < A < 2^w$ et dad est l'opérateur de «*décalage à droite*».



Méthode de multiplication

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Une autre façon de définir $h(k)$:

$$h(k) = (A \cdot k \bmod 2^w) \text{ dad}(w - r)$$

où A est un *entier impair* tel que $2^{w-1} < A < 2^w$ et dad est l'opérateur de «*décalage à droite*».

- ne pas prendre A trop proche de 2^w



Méthode de multiplication

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Une autre façon de définir $h(k)$:

$$h(k) = (A \cdot k \bmod 2^w) \text{ dad}(w - r)$$

où A est un *entier impair* tel que $2^{w-1} < A < 2^w$ et dad est l'opérateur de «*décalage à droite*».

- ne pas prendre A trop proche de 2^w
- la multiplication modulo 2^w est rapide



Méthode de multiplication

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

- Une autre façon de définir $h(k)$:

$$h(k) = (A \cdot k \bmod 2^w) \text{ dad}(w - r)$$

où A est un *entier impair* tel que $2^{w-1} < A < 2^w$ et dad est l'opérateur de «*décalage à droite*».

- ne pas prendre A trop proche de 2^w
- la multiplication modulo 2^w est rapide
- l'opérateur dad est rapide.



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(k) = (A \cdot k \bmod 2^w) \bmod (w - r)$$



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(k) = (A \cdot k \bmod 2^w) \bmod (w - r)$$

- Supposons que $m = 8 = 2^3$ et que l'ordinateur travaille avec des mots de 7 bits.



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Wegrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(k) = (A \cdot k \bmod 2^w) \bmod (w - r)$$

- Supposons que $m = 8 = 2^3$ et que l'ordinateur travaille avec des mots de 7 bits.
- Prenons $A = 1011001_{(2)}$ et $k = 1101011_{(2)}$



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(k) = (A \cdot k \bmod 2^w) \bmod (w - r)$$

- Supposons que $m = 8 = 2^3$ et que l'ordinateur travaille avec des mots de 7 bits.
- Prenons $A = 1011001_{(2)}$ et $k = 1101011_{(2)}$
- Le produit $A \cdot k$

$$\begin{array}{rcl} & 1011001 & = A \\ \times & 1101011 & = k \\ \hline 1001010 & 0110011 & = A \cdot k \\ & A \cdot k \bmod 2^7 & \end{array}$$



Exemple

Les tables de
hachage

Nour-Eddine
Oussous, Éric
Węgrzynowski

Plan

Introduction

Proposition de
solution

Solution :
Table de
hachage

$$h(k) = (A \cdot k \bmod 2^w) \text{ dad}(w - r)$$

- Supposons que $m = 8 = 2^3$ et que l'ordinateur travaille avec des mots de 7 bits.
- Prenons $A = 1011001_{(2)}$ et $k = 1101011_{(2)}$
- Le produit $A \cdot k$

$$\begin{array}{rcl} & 1011001 & = A \\ \times & 1101011 & = k \\ \hline 1001010 & 0110011 & = A \cdot k \\ & A \cdot k \bmod 2^7 & \end{array}$$

- $w - r = 7 - 3 = 4$. On fait un décalage à droite de 4 bits :
 $h(k) = 011$.