Chapitre 12 – Table d’association

Une table d’association est un type de données associant à un ensemble de clefs, un ensemble correspondant de valeurs. Chaque clef est associée à une seule valeur.

Une table d’association permet de stocker des couples [clés, valeurs].

Problème et idée

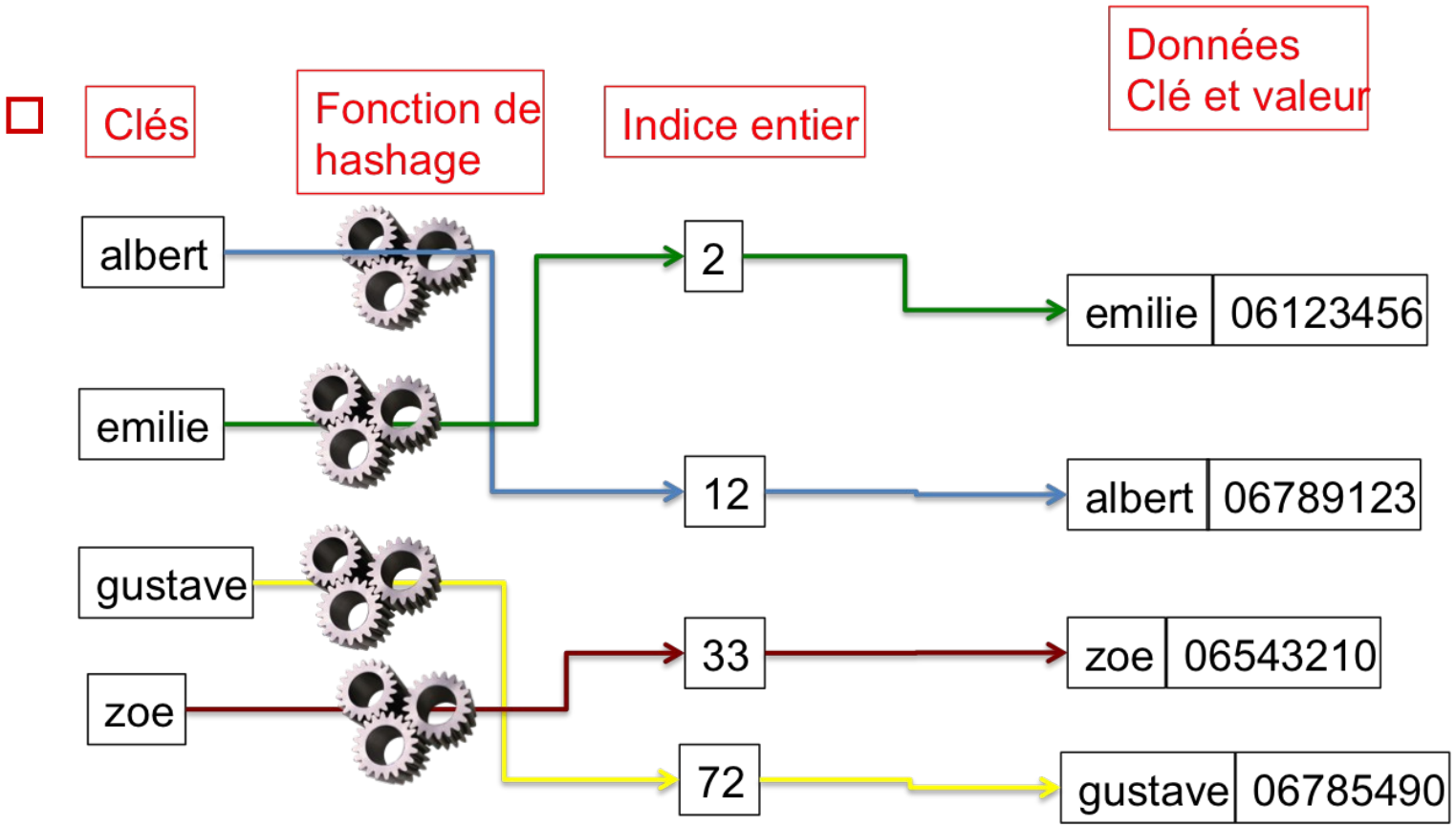
- La recherche d'un élément dans un tableau ou une liste dépend du nombre d'éléments n
  O(n) dans le pire des cas (n comparaisons)

- La recherche d'un élément dans une table d'association ne dépend pas du nombre d'éléments n
  O(1) accès direct

Hachage

- Le premier principe est de transformer une clé en un entier, qui sera alors l'indice du tableau contenant les données.
- Le hachage assure une correspondance entre notre clé et l'indice entier du tableau.
  Recherche en O(1)

Hachage


Clés
Fonction de hashage
Indice entier
Données
Clé et valeur
albert
emilie
gustave
zoe
2
12
33
72
emilie 06123456
albert 06789123
zoe 06543210
gustave 06785490

Fonction de hachage : propriétés

- Rapidité de calcul
- Dispersion : répartir au mieux toutes les valeurs sur l'ensemble des entiers autorisés [0..M-1]
- Doit minimiser
  - les indices inutilisés (espaces mémoires)
  - les collisions (différentes clés donnent la même valeur d'indice)
- Facteur de remplissage : $\alpha = n / M$

Fonction de hachage : exemples

- Clef k de type chaine
  h(k) = somme(rang de chaque car de k) mod M

  h(k) = somme((ASCII de chaque car de k) * 256^i) i ϵ [0,n-1]
- Clef k de type entier
  h(k) = k mod M
  h(k) = [M * ((a*k) mod card(clef)) / card(clef)], a entier fixé
- Clef k de type réel
  h(k) = partie_entière (k) mod M

Collision

- Se produit chaque fois que la fonction de hachage donne un même indice pour 2 clés différentes
  par exemple : ‘male’ et ‘lame’
- Sont dues au fait qu'aucune fonction de hachage n'est une injection de l'espace des clés vers l'espace 0..M-1
- Le comportement dans le pire cas est en O(n) si tous les éléments sont dans la même case
- Si un élément a une chance équitable de se retrouver dans une des cases, le hachage est *uniforme simple*

TAD Table d'association

Sorte : Table-assoc

Utilise : Clef, Valeur, Entier
[Clef et Valeur peuvent être de types chaines, entier, ...]

Opérations :

Créer-table-vide(Entier) → Table-assoc	{constructeur}
Insérer(Table-assoc, Clef, Valeur) → Table-assoc	{constructeur}
Supprimer(Table-assoc, Clef) → Table-assoc	{constructeur}
Chercher(Clef) → Valeur	{observateur}

Hachage par liste

- La table est une séquence de M listes (Seq de Liste)
- Chaque élément de la liste est un couple [clef, valeur]
- Les insertions dans la liste se font en tête de liste
- Pour les suppressions, on cherche la place de l'élément à supprimer dans la liste (chercher-elt) et on la supprime
- Solution au problème des collisions

Hachage ouvert

- La table est une séquence de M enregistrements.
- Si collision, on cherche une case vide parmi les cases suivantes à l'aide d'une fonction de sondage (linéaire, quadratique, double hachage,…)
- Pour la suppression, on marque la case qui contenait la clef supprimée avec la clef spéciale SUP
- Pour l'insertion, si on trouve une clef SUP, on considère cette case comme disponible
- Pour la recherche, si on trouve une clé SUP, on considère cette case comme occupée.

Complexité

On suppose la fonction de hachage uniforme

	Insertion	Recherche	Suppression
Hachage par liste	O(1)	O(α)	O(α)
Hachage ouvert sondage linéaire	O(1 / (1-α))	O(1 / (1-α))	O(1 / (1-α))

Au pire des cas, le nombre de sondages vaut au plus $1 / (1 - \alpha)$ avec une fonction de hachage uniforme

Conclusion

Hachage par chaînage

- Nombre illimité de clés/valeurs et de collisions
- Performances stables
- Coût de la gestion des listes

Adressage ouvert

- Suppression avec gestion de la clef spéciale SUP
- Rapide et peu coûteux en mémoire
- Nombre de clés/valeurs limité à la taille de la table.