NFP136- VARI 8

LE HACHAGE

TABLES DE HACHAGE

Définition

- **Structure de données reposant sur des tableaux**
- **Utilisées pour stocker de grandes quantités d'informations**
- ***Recherche rapide d'un élément***
- **Tableau [0,…,m-1] Indice dans la table calculé en fonction de la clé de l'élément stocké à cet endroit**

PRINCIPE DU HACHAGE

Fonction de hachage : h

h : {clés} ⟶ [0,..,m-1]

c ⟶ h(c) ***accès direct dans la table***

Exemple

Dans un annuaire téléphonique, on veut pouvoir retrouver l'information **nom, prénom** à partir de la clé **numéro de téléphone**

Fonction de hachage (on ne dit pas ici comment on la calcule):

h(0381111144)=0 ; h(0381333333)=2 ; h(0381222222)=3 ; h(0381123456)=6

Exemple

Dans un annuaire téléphonique, on veut pouvoir retrouver l'information **nom, prénom** à partir de la clé **numéro de téléphone**

Fonction de hachage (**on ne dit pas ici comment on la calcule**):

h(0381111144)=0 ; h(0381333333)=2 ; h(0381222222)=3 ; h(0381123456)=6

Avantage, accès direct dès lors qu'on a calculé h(c)

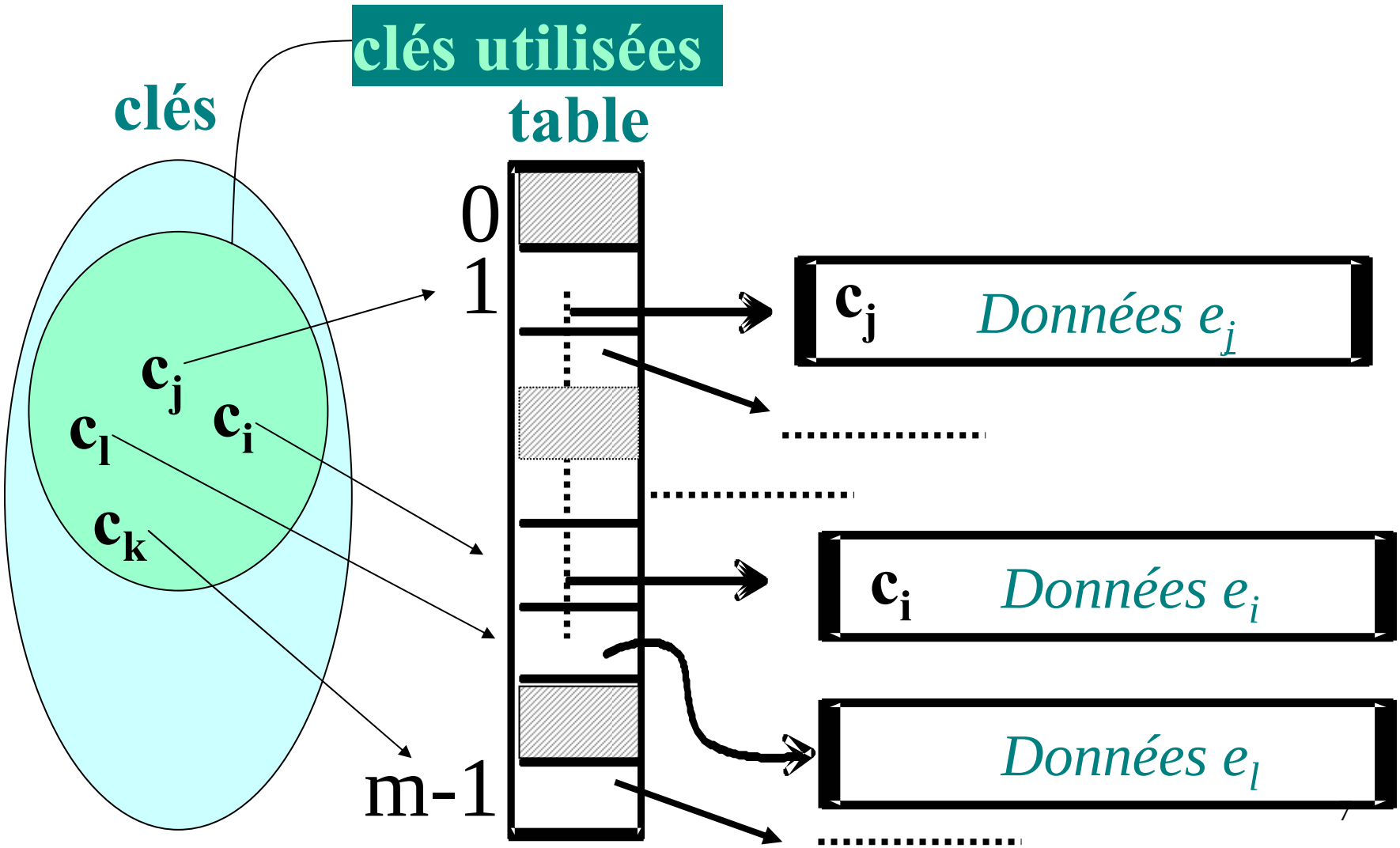
Évite d'avoir un tableau de taille = tous les numéros à 10 chiffres !

Table de hachage:

indice	information
0	Pierre Durand 0381111144
1	
2	Paul Dupont 0381333333
3	Yvette Bon 0381222222
4	
5	
6	Gilles Dupont 038123456

Cas simple: adressage direct

Fonction h injective: une clé ⟶ une place



Mais le cas simple n'est pas réaliste

- **En général, h non injective:**

on peut avoir $c \neq c'$ et $h(c) = h(c')$

On parle de COLLISION

COLLISION

h non injective

E = {éléments}={e_1, e_2,..,e_i,..,e_n}

E ⟶ {clés} ⟶ [1,2,..,n]

e_i ⟶ clé(e_i) ⟶ h(clé(e_i))

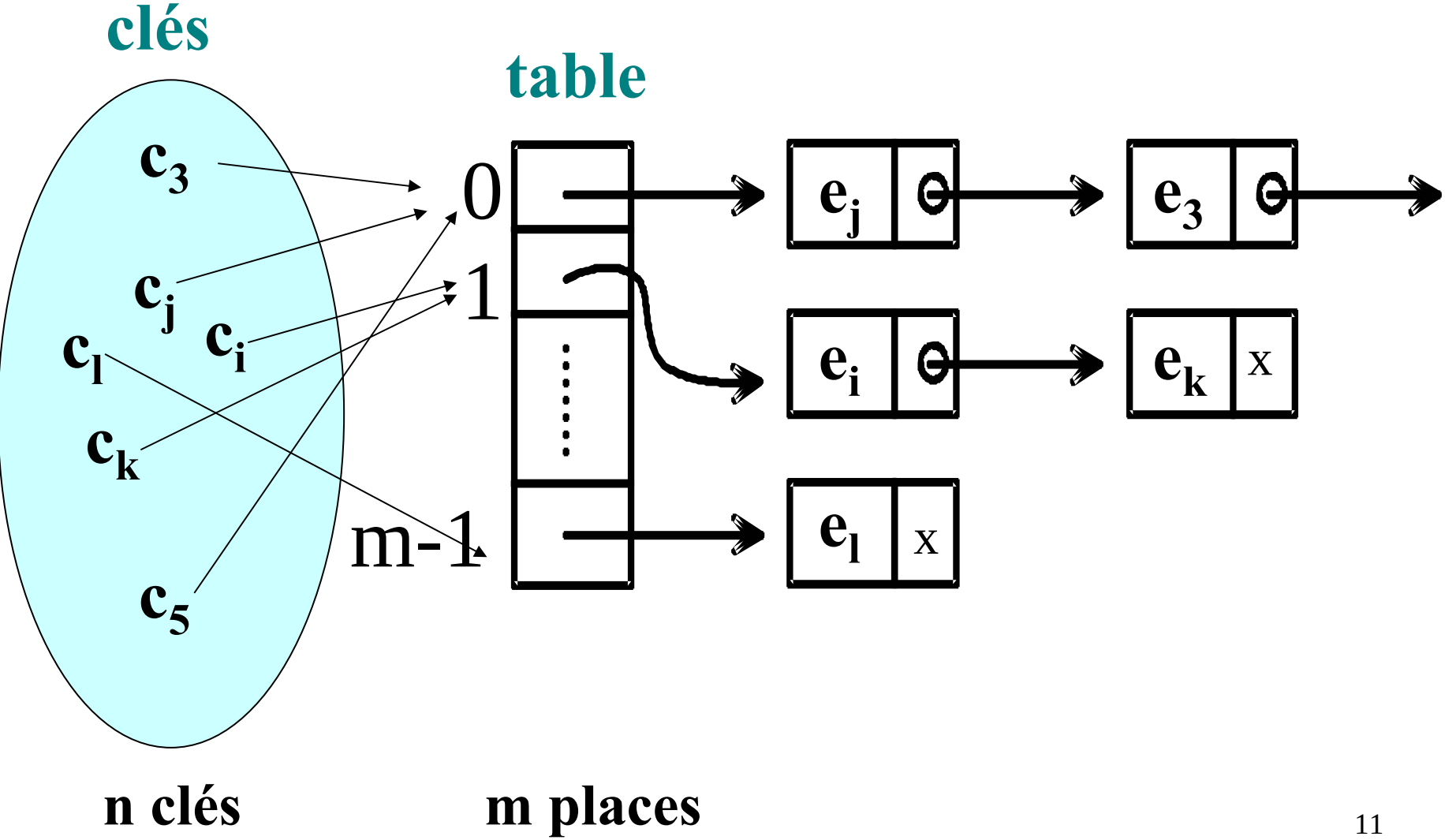
Place de l'élément e_i : T[h(clé(e_i)]

Collision si h(clé(e_i)) = h(clé(e_j))

Gestion des collisions
1ère méthode : chaînage

Principe : Les clefs qui ont la même valeur de hachage sont mises dans la même liste

adressage indirect et chaînage



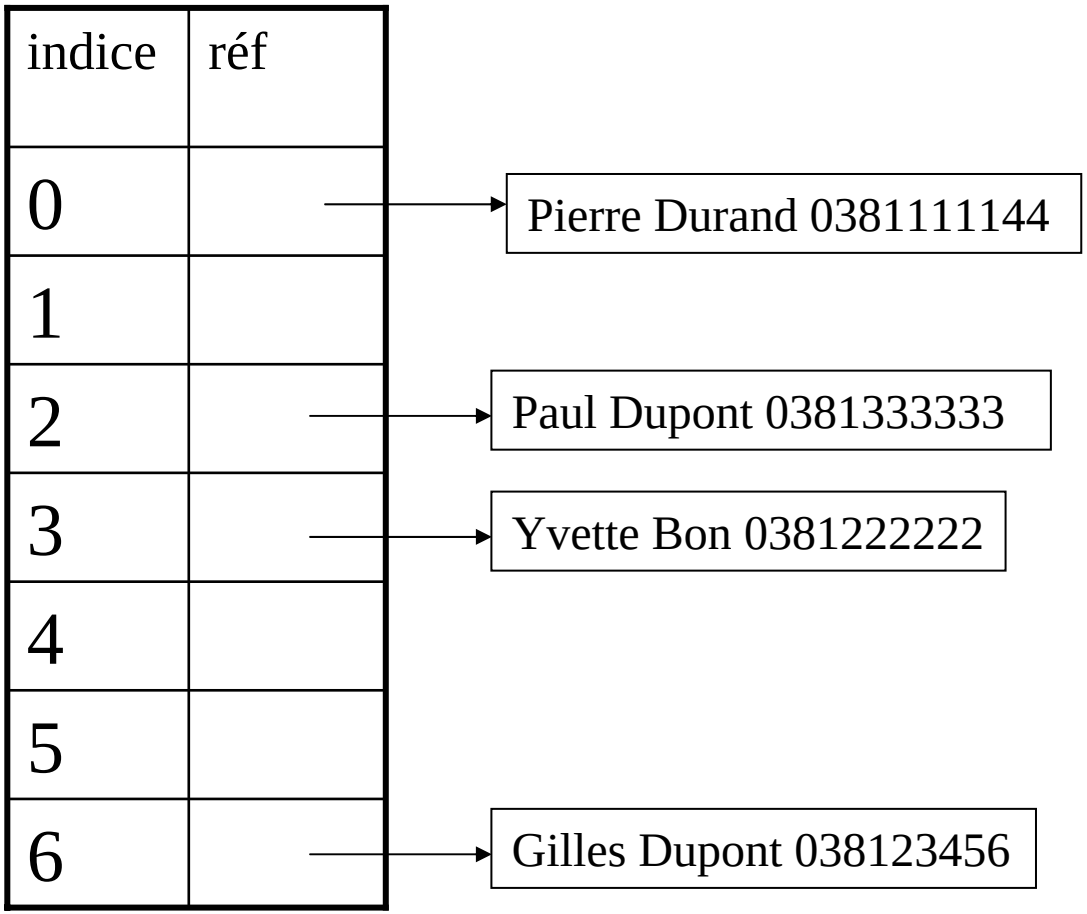
Exemple

Table de hachage:

indice	information
0	Pierre Durand 0381111144
1	
2	Paul Dupont 0381333333
3	Yvette Bon 0381222222
4	
5	
6	Gilles Dupont 038123456

Exemple

Table de hachage **devient** :



Exemple

Si on doit ajouter : Alain Bernard, 0140272000 et que h(0140272000)=3

- **Temps de recherche de l'élément de clé c**

  =

  Calcul de l'indice h(c) dans T: O(1) +

  + Parcours de la liste chaînée : $O(L_{h(c)})$

- **Idéal: éviter les collisions**
- **Réaliste: limiter les collisions**

- **Important: choisir une « bonne » fonction de hachage**

Gestion des collisions
2ème méthode : adressage ouvert

Principe (le plus simple) : On cherche la première place libre dans le tableau après la valeur de hachage

Exemple

Table de hachage:

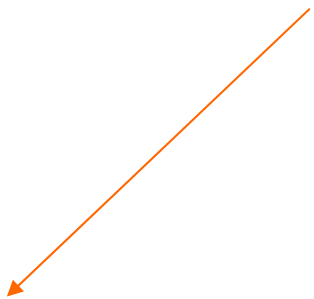
indice	information
0	Pierre Durand 0381111144
1	
2	Paul Dupont 0381333333
3	Yvette Bon 0381222222
4	
5	
6	Gilles Dupont 038123456

Exemple

Table de hachage:

indice	information
0	Pierre Durand 0381111144
1	
2	Paul Dupont 0381333333
3	Yvette Bon 0381222222
4	**Alain Bernard 0140272000**
5	
6	Gilles Dupont 038123456

Si on doit ajouter : Alain Bernard, 0140272000 et que h(0140272000)=3



- **Prévoir une table de taille plus importante**
- **Pour rechercher un élément de clé c, calculer h(c) puis regarder dans T[h(c)] et tous les indices suivants jusqu'à trouver c**
- **Insertion et recherche en O(n) où n est la taille du tableau**
- **On peut faire mieux en n'envisageant pas tous les éléments consécutifs**

FONCTIONS DE HACHAGE DE BASE

Une fonction de hachage associe un entier à un objet (une chaîne, un entier, ou un objet quelconque)

4 règles d'or (irréalisables)

1. La valeur de hachage ne dépend que de l'objet d'entrée
2. La fonction de hachage utilise tous les champs de l'objet d'entrée
3. La fonction de hachage distribue uniformément les données
4. La fonction de hachage génère des valeurs très différentes pour des objets quasi identiques

De plus, la fonction de hachage doit être rapide à calculer

Un exemple

Les 4 règles sont-elles vérifiées ?

```
static int codeHachage(String s, int m){
                        //une mauvaise fonction de H
  int sum = 0;
  for (int i = 0; i < s.length(); i++) {
    char c = s.charAt(i);
    sum =sum + c;
  }
  return sum % m;
}
```

Mauvaise fonction de hachage : “Bela Atea” et “Lea Abate” donnent la même sum et donc ont le même H code

Fonctions de hachage pour les entiers

- h(k) le reste de la division de k par m : h(k) = k mod m

Une bonne valeur pour m est un nombre premier (pas trop proche d'une puissance de 2). Peut se comporter très mal quand même.

- Variante de la division h(k) = k(k + 3) mod m (Knuth)

- Multiplication . $m = 2^r$ et A un réel $h(k) = \lfloor m(Ak - \lfloor Ak \rfloor) \rfloor$

Knuth propose le nombre d'or $A = \frac{1}{2}(\sqrt{5} - 1)$

Fonctions de hachage pour les String

- Partir du code ASCII de chaque caractère (entier sur 7 bits, compris entre 0 et 127)

- H code d'une chaîne de caractères = entier en base 128

- Exemple Le H code de la chaîne "java" est :

$106 \times 128^3 + 97 \times 128^2 + 118 \times 128 + 97 = 223902561$