

## TITRE LONG

Jusqu'ici, nous avons utilisé des structures de données basées sur les comparaisons entre les données. Nous allons dorénavant étudier le hachage qui se caractérise par un calcul externe.

Le hachage consiste à calculer à l'aide d'une "fonction de hachage" la position  $h(d)$  de chaque donnée  $d$ . Un ensemble de données  $x_1, \dots, x_n$  sera représenté en stockant  $x_i$  à l'adresse  $h(x_i)$ .

### Méthode d'accès direct

Si les configurations particulières des données font que  $\forall x_i \neq x_j, h(x_i) \neq h(x_j)$ , alors on dispose d'une méthode d'accès à toutes les opérations de base :

- Ajouter, insérer
- Supprimer
- Réduire

Coûteront  $O(1)$ , le coût du calcul de  $h$ .

### Méthode de résolution de conflits

Lorsqu'une ou plusieurs clés partagent une même position, il y a collision. La méthode de hachage diffère des autres par sa méthode de résolution des conflits.

### Hachage avec chaînage

On traite les conflits en stockant les clés dans des zones séparées tel des liste chaînées.

Dans ce cas, on a :

Opération	Coût
Insertion	$O(1)$ (insertion en tête de liste)
Suppression	$O(1)$ dans le cas idéal, $O(1 + \alpha)$ sinon
Recherche	$O(1 + \alpha)$

Analyse du hachage :

**Hypothèses :** Chaque clé a la même probabilité d'être hachée dans n'importe quelle liste (hachage uniforme). Par ailleurs on dispose d'une table de hachage  $T$  avec  $n$  positions et stockant  $n$  clés.

Soit  $\alpha = \frac{n}{m}$ .

Le calcul  $h(k)$  est en  $O(1)$ .

Pour rechercher un élément, on calcule sa position  $h(k)$ , puis on cherche l'élément de la liste stockée à cette position. Pour calculer le coût de l'opération de recherche, on distingue 2 cas :

- 1<sup>er</sup> cas : Recherche infructueuse, l'élément ne se trouve pas dans T. Le coût est  $O(1) + O(\alpha) = O(1 + \alpha)$ .
- 2<sup>ème</sup> cas : Recherche avec succès. Le coût est lui aussi en  $O(1 + \alpha)$ .

### Hachage avec adressage ouvert

On cherche un élément dans la première case libre. Il n'y a pas nécessité de liste, de pointeur, pas de structure de données extérieure additionnelle. Un sondage produit une séquence de la forme  $h(k, 0); h(k, 1) \dots$ .

```
HachageAdressageOuvert_Search(T,k)
```

```
  i <- 0
```

```
  Répéter
```

```
    j <- h(k,i)
```

```
    Si T[j] = k Alors
```

```
      retourner j
```

```
    Finsi
```

```
  Jusqu'à T[j] = NULL ou i = m
```

```
  retourner NULL
```

```
Fin
```

### Choix de la fonction de hachage

Cette problématique sera vue lors des séances de TD.