

TD 4 - Tables de hachage

L'objectif de ce TD est de stocker les informations relatives à la liste de personnes ci-contre (étudiants, clients, ...), identifiés par un numéro unique.

Dans tout ce TD, on utilise la fonction de hachage qui calcule le reste modulo 9 d'un entier, et $B = 9$.

Par simplicité, on ne stocke que le prénom, mais dans une application plus réaliste on aurait une structure avec nom, adresse e-mail, etc.

Clé	Élément
233	David
311	Matt
674	Chris
649	Peter
452	Jodie
152	Tom
324	Jon
963	Bill

Exercice 1. Fonction de hachage

Calculer la valeur de hachage pour chaque élément.

Exercice 2. Hachage fermé

1. Calculer la table de hachage fermée obtenue après insertion des éléments ci-dessus, dans l'ordre indiqué.
2. En comptant 1 opération pour calculer une valeur de hachage et 1 opération pour comparer deux clés, combien d'opérations sont nécessaires pour créer la table ? (On suppose que pour chaque insertion, on vérifie au préalable que la clé est effectivement nouvelle.)
3. Combien d'opérations auraient été nécessaires en utilisant une liste chaînée classique ?

Exercice 3. Implémentation du hachage fermé

Implémenter les fonctions de manipulation sur les tables de hachage fermé (penser à maintenir `size` à jour) :

```
value_t get(Hashtable* t, key_t k);  
void set(Hashtable* t, key_t k, value_t v);  
void delete(Hashtable* t, key_t k);
```

Pour `get`, on utilisera la valeur `null` si la clé n'est pas présente.

`set` peut soit insérer une nouvelle valeur si la clé n'était pas présente, soit modifier la valeur existante.

À terminer en TP

Exercice 4. Hachage ouvert

1. Calculer la table de hachage ouvert obtenue après insertion des éléments ci-dessus, dans l'ordre indiqué.
2. En comptant 1 opération pour calculer une valeur de hachage et 1 opération pour comparer deux clés, combien d'opérations sont nécessaires pour créer la table ?

Exercice 5. Recherche et Suppression

Dans la table calculée précédemment, décrire les comparaisons de clés et les modifications effectuées lors des opérations suivantes :

1. Recherche de la clé **860**.
2. Recherche de la clé **452**.
3. Suppression de la clé **649**.
4. Recherche de la clé **324**.

Exercice 6. Implémentation du hachage ouvert

Implémenter les fonctions de manipulation sur les tables de hachage ouvert :

```
value_t get(Hashtable* t, key_t k);  
void set(Hashtable* t, key_t k, value_t v);  
void delete(Hashtable* t, key_t k);
```

À terminer en TP