

NOTE: ce document est une ébauche de documentation.

En informatique, une **table de hachage** est une **structure de données** qui permet l'association de **paires clé/valeur**. La clé étant une chaîne de caractère, la valeur un nombre, ou une suite de nombres comme un numéro de téléphone ou n'importe quelle donnée associée à une "valeur".

Principe de fonctionnement : on accède à chaque élément de la table via sa **clé**. Il s'agit d'un [tableau](#) ne comportant pas d'ordre (un tableau est indexé par des [entiers](#)). Les paires sont stockées dans les alvéoles du tableau. En fonction de la méthode utilisée, le remplissage peut poser problème ou pas.

Les opérations que l'on peut réaliser avec une table de hachage sont :

- créer la table (et l'initialiser) ;
- ajouter une paire clé/valeur ;
- supprimer une paire en donnant la clé (la valeur sera supprimée en même temps que la clé) ;
- chercher si une clé existe dans la table ;
- retrouver la valeur associée à une clé ;
- modifier la valeur associée à une clé.

La taille de la table est **a priori** indéfinie (mais de valeur finie), et dépend des ressources mémoire dont on dispose.

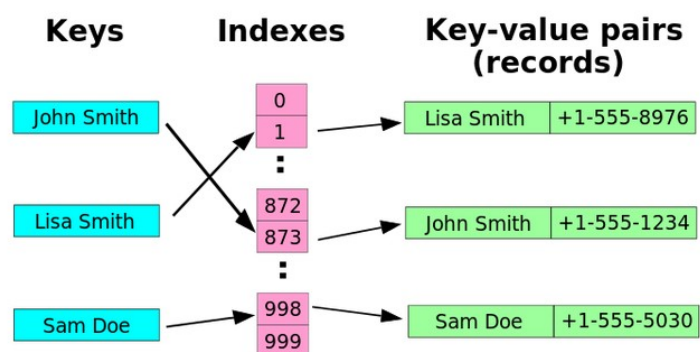
Il n'y a pas de relation d'ordre dans une table de hachage, et la position des paires clé-valeur est pseudo-aléatoire dans cette table. Cette structure n'est donc pas adaptée au feuilletage (*browsing*) de données voisines. Des types de structures de données comme les [arbres équilibrés](#) [4], généralement plus lents (en $O(\log n)$ [3]) et un peu plus complexes à implémenter, maintiennent une structure ordonnée.

On associe la notion de **dictionnaire** (très utilisé avec Mac OSX) à la notion de table de hachage.

Exemple de table de hachage

Pour illustrer la notion de table de hachage, on utilise l'exemple donné sur wikipedia [1], cf figure ci-contre.

Les données saisies par l'utilisateur sont en bleu et en vert. L'indice associé à l'index est attribué de façon interne par le logiciel, et représente la position dans la liste chaînée qui n'est pas ordonnée.



Dans le TP, l'indice sera égal à la valeur de refcount (très utile en C++ aussi) dans le code écrit en langage C.

Remarque : ne pas confondre table de hachage avec fonction de hachage : On nomme **fonction de**

hachage, de l'anglais *hash function* (*hash* : pagaille, désordre, recouper et mélanger) par analogie avec la cuisine, une [fonction](#) particulière qui, à partir d'une donnée fournie en entrée, calcule une ***empreinte numérique*** servant à identifier rapidement la donnée initiale, au même titre qu'une signature pour identifier une personne. Les fonctions de hachage sont utilisées en [informatique](#) et en [cryptographie](#) notamment pour reconnaître rapidement des fichiers ou des mots de passe.

Les deux notions sont toutefois proches, puisqu'on réalise une opération sur la clé pour obtenir la valeur

Analyse du code

Organisation de l'arborescence

Travail à effectuer pour aller plus loin

Bibliographie

Sources type URL :

[1] https://fr.wikipedia.org/wiki/Table_de_hachage

[2] https://fr.wikipedia.org/wiki/Fonction_de_hachage

[3] https://fr.wikipedia.org/wiki/Comparaison_asymptotique

[4] https://fr.wikipedia.org/wiki/Arbre_équilibré

La page du département informatique de l'ENS Cachan (bouh, Paris Saclay maintenant) :

<http://www.dptinfo.ens-cachan.fr/Agregation/>

Document de Madame Cristina Sirangelo :

<http://www.dptinfo.ens-cachan.fr/Agregation/Algo13-14/3-tables-hachage.pdf>

<http://www.lsv.fr/~jacomme/agreg/index.html> : fiches d'algorithmique

Les dictionnaires (Université de Montréal) :

<https://www.iro.umontreal.ca/~hamelsyl/Dictionnaires4.pdf>

Le site de la préparation à l'Agrégation de Mathématiques, option informatique :

<https://charlie-j.github.io/agreg-fiches/>

Les sources : <https://github.com/charlie-j/agreg-fiches>