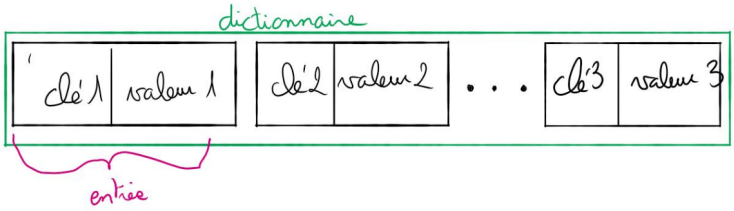
Les dictionnaires

Table des matières

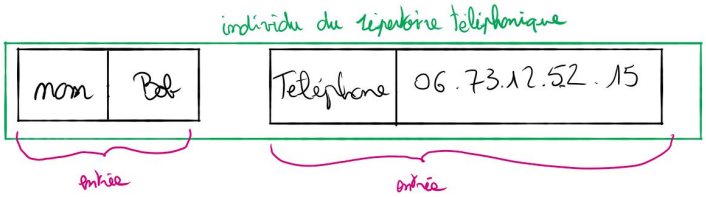
1) Les dictionnaires

1.1) Définition

Un dictionnaire (aussi appelé tableau associatif) est une structure de données qui permet d'associer une valeur à une clé. Cette clé peut être un mot ou un entier. l'ensemble clé-valeur est appelé entrée.



Exemple



1.2) Interface des dictionnaires

Voici les opérations que l'on peut effectuer sur le type abstrait dictionnaire :
- *ajout*() : on associe une nouvelle valeur à une nouvelle clé
- *modif*() : on modifie un couple clé :valeur en remplaçant la valeur courante par une autre valeur (la clé restant identique)
- *suppr*() on supprime une clé (et donc la valeur qui lui est associée)
- *rech*() on recherche une valeur à l'aide de la clé associée à cette valeur.

Exemples : Soit le dictionnaire D composé des couples clé :valeur suivants => prenom :Kevin, nom :Durand, date-naissance :17-05-2005. Pour chaque exemple ci-dessous on repart du dictionnaire d'origine :

- $ajout(D, tel$: 06060606) ; le dictionnaire D est maintenant composé des couples suivants : prenom :Kevin, nom :Durand, date-naissance :17-05-2005, tel :06060606
- $modif(D, nom : Dupont)$; le dictionnaire D est maintenant composé des couples suivants : prenom :Kevin, nom :Dupont, date-naissance :17-05-2005
- $suppr(D, date - naissance)$; le dictionnaire D est maintenant composé des couples suivants : prenom :Kevin, nom :Durand
- $rech(D, prenom)$; la fonction retourne Kevin

1.3) Implémentation

L'implémentation des dictionnaires dans les langages de programmation peut se faire à l'aide des tables de hachage. Les tables de hachages ainsi que les fonctions de hachages qui sont utilisées pour construire les tables de hachages, ne sont pas au programme de NSI. Cependant, l'utilisation des fonctions de hachages est omniprésente en informatique, il serait donc bon, pour votre "culture générale informatique", de connaitre le principe des fonctions de hachages. Voici un texte qui vous permettra de comprendre le principe des fonctions de hachages : c'est quoi le hachage . Pour avoir quelques idées sur le principe des tables de hachages, je vous recommande le visionnage de cette vidéo : wandida : les tables de hachage

Si vous avez visionné la vidéo de wandida, vous avez déjà compris que l'algorithme de recherche dans une table de hachage a une complexité O(1) (le temps de recherche ne dépend pas du nombre d'éléments présents dans la table de hachage), alors que la complexité de l'algorithme de recherche dans un tableau non trié est O(n). Comme l'implémentation des dictionnaires s'appuie sur les tables de hachage, on peut dire que l'algorithme de recherche d'un élément dans un dictionnaire a une complexité O(1) alors que l'algorithme de recherche d'un élément dans un tableau non trié a une complexité O(n).

2) Exercices

Exercice 1 :

Préciser quelle est la structure de donnée à privilégier pour chacune de ces taches :

1. Représenter un répertoire téléphonique.
2. Stocker l'historique des actions effectuées dans un logiciel et disposer d'une commande Annuler.
3. Envoyer des fichiers au serveur impression.
4. On souhaite stocker un texte très long que l'on souhaite pouvoir modifier

Exercice 2 :

Dans chacun des cas suivants, déterminez quelle structure de données est la plus adaptée :

1. Gérer le flux des personnes arrivant à la caisse d'allocations familiales ;
2. Mise en place d'un mécanisme annuler/refaire pour un traitement de texte ;
3. Garder les nombres premiers parmi les nombres de 2 à 1000 en utilisant la méthode du crible d'Eratosthène.
4. Lors d'une partie échec, on veut enregistrer l'ensemble des coups joués et pouvoir les consulter.