

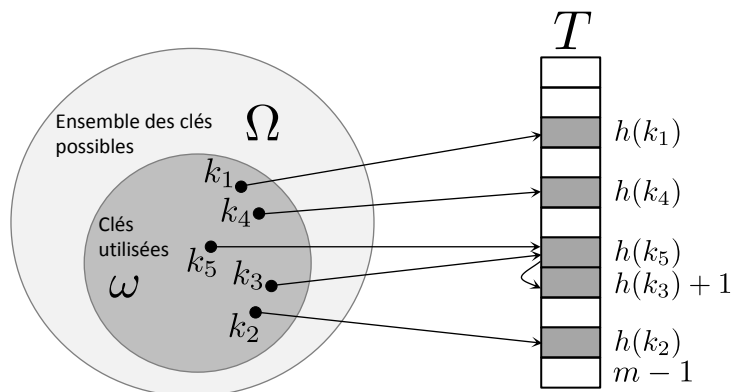
Algorithmique - TD6

Table de hachage à adressage ouvert

1 Introduction

Dans ce TP, vous allez concevoir et implémenter en langage C une table de hachage avec adressage ouvert.

Voici pour rappel le fonctionnement d'une telle table :



Lorsque la valeur de la fonction de hachage donne deux fois la même valeur, il faut utiliser une autre case du tableau (d'où l'aspect ouvert de l'adressage). Vous utiliserez la fonction de codage et de hachage de votre choix sachant que nous cherchons à stocker des chaînes de caractères pour les clés et les valeurs.

2 Travail à effectuer

- Donner la structure de données qui va permettre de représenter une seule cellule de la table. Celle-ci contiendra trois informations : le statut de la cellule (vide, remplie, ou supprimée), la clé de la cellule dans le cas où elle est remplie (chaîne de caractères) et la valeur associée à la clé (chaîne de caractères).
- Donner la structure de données qui va permettre de représenter la table de hachage.
- Donner les algorithmes (pré et post-conditions, entrées et sorties) des opérations suivantes :
 - Initialiser la table (argument : la taille de la table sous forme d'entier)
 - Insérer un élément à partir du couple (clé, valeur). Cette opération écrase la valeur associée si la clé était déjà présente dans le dictionnaire.
 - Supprimer un élément à partir de sa clé
 - Effectuer une requête sur une clé
 - Afficher les statistiques de la table : nombre total de cases, nombre de cases remplies, supprimées et vides.
 - Libérer la structure de données
- Implémenter ces algorithmes en langage C sur la plateforme de TD.

3 Description des entrées/sorties

Pour valider votre solution, les entrées et les sorties doivent respecter un format très précis. Les entrées possibles sont :

```
init <n>
insert <cle> <valeur>
delete <cle>
query <cle>
stats
destroy
bye
```

La commande `init` permet d'allouer la mémoire nécessaire à la table de hachage. Elle prend en argument le nombre de cases du tableau `<n>` un nombre entier. `<cle>` et `<valeur>` représentent des chaînes de caractères quelconques sans espace. La commande `insert` permet d'ajouter une paire (clé, valeur) au dictionnaire. La commande `delete` permet de supprimer un élément du dictionnaire. La commande `query` permet de faire une requête sur une clé. Si cette clé est présente, la valeur associée sera affichée, si elle ne l'est pas un message spécial sera affiché. La commande `destroy` permet de libérer la mémoire. La commande `bye` termine l'exécution du programme, et la commande `stats` permet d'afficher les statistiques de la table (voir ci-dessous pour le format de sortie, attention à bien respecter le nombre d'espaces, ceux-ci ont été symbolisés par un symbole spécial dans le format de sortie). Chaque ligne affichée sera terminée par deux caractères (CR et LF), correspondant à la chaîne `"\r\n"`.

Un exemple d'entrée :

```
init 100
insert a b
insert a c
query a
query b
insert b d
delete a
query a
stats
destroy
bye
```

avec la sortie correspondante :

```
c
Not_found
Not_found
size_:::::100
empty_:::98
deleted_:1
used_:::::1
```

Une ossature de programme en langage C vous est fournie sur moodle, elle permet l'interprétation des commandes décrites ci-dessus et vous donne une fonction de hachage efficace pour une chaîne de caractères.