

TD3 – Maps et tables de hachage

ING1 – Algorithmique II

Année 2011–2012



1 Opérations élémentaires

- ① Écrire un programme qui affiche le nom de tous les étudiants ainsi que leur moyenne en utilisant la map `mapMoyennes: Map<chaîne> de réel` vue en cours. Vous afficherez également la moyenne générale. On considèrera la procédure `calculMoyenne(mapMoyennes: Map<chaîne> de réel (E/S))` qui prend en paramètre une map préalablement initialisée et qui la remplit avec le nom des étudiants associé à leur moyenne.

- ② **Recherche de l'élément minimum**
Écrire une méthode qui recherche la valeur minimale dans une map d'entiers (type de clef quelconque) et qui retourne sa clef et sa valeur.

2 Fonctions de hachage

- ③ **Méthode dite de la multiplication**
On considère une table de hachage de taille $m = 1000$ et la fonction de hachage $h(k) = \lfloor m \times (kA - \lfloor kA \rfloor) \rfloor$ pour $A = \frac{\sqrt{5}-1}{2}$. Écrire la fonction de hachage. Faire un jeu d'essai pour les clefs 61, 62, 63, 64 et 65.

Dans toute la suite du TD, on considère qu'il existe une procédure *convertir* qui prend en paramètre une chaîne de caractères (correspondant à un numéro de téléphone français sans espace) et un tableau d'entiers. Cette procédure met chaque chiffre du numéro de téléphone dans le tableau ; par exemple, "0559059090" deviendrait {0,5,5,9,0,5,9,0,9,0}.

- ④ Écrire une fonction de hachage qui calcule la valeur de hachage d'un numéro de téléphone en faisant la somme des 10 chiffres modulo 10. Faire quelques jeux d'essai. Quel(s) problème(s) rencontrez-vous ?

3 Tables de hachage et annuaire inversé

On considère un annuaire inversé stocké à l'aide d'une table de hachage. La table de hachage est un tableau de listes de *Contact* dont la structure est la suivante :

type Contact = structure

téléphone: chaîne

nomPrénom: chaîne

fin type

On utilisera la fonction de hachage définie dans la question 4 afin de transformer la clef (qui correspond donc au numéro de téléphone) en indice. Cet indice permet d'accéder, via la table de hachage, à la liste des contacts ayant la même valeur de hachage.

- 5 | **Ajout d'un numéro dans l'annuaire**
Écrire une méthode qui permet d'ajouter un nouveau contact dans l'annuaire inversé.
- 6 | **Recherche d'un numéro dans une liste de contacts**
Écrire une méthode qui permet de rechercher un propriétaire dans une **liste** de contacts à partir de son numéro de téléphone, et qui retourne soit le nom et prénom du propriétaire s'il est présent, soit une chaîne vide.
- 7 | **Recherche d'un numéro dans l'annuaire**
Écrire une méthode qui recherche un numéro dans l'**annuaire** et qui retourne soit le nom et prénom du propriétaire s'il est présent, soit une chaîne vide.
- 8 | **Complexité de la recherche**
1. Déterminer la complexité de la méthode de recherche.
 2. Quelle est la complexité de la méthode de recherche si on change le modulo 10 en modulo 50 dans la fonction de hachage? En modulo 90?
 3. Quelle serait la complexité de la méthode de recherche en adressage direct?

9

En utilisant la table de hachage définie précédemment, écrire les méthodes de base de la map :

1. cardinalité
2. stocker
3. valeurDe
4. supprimer
5. valeurExiste
6. clefExiste

□