GIS2A3 — Structures de Données — Épreuve de Mai 2017
Durée 2 heures — Aucun document autorisé

---

1 Petits Exercices (4 points)

On considère l'insertion des clés $\square$, $\bigcirc$, $\triangle$, $*$ dans une table de hachage de $N = 12$ alvéoles. La table est gérée avec la technique du double hachage. Les valeurs de hachage des différents éléments sont les suivantes :

	$\square$	$\bigcirc$	$\triangle$	$*$
h_1	3	7	7	4
h_2	5	7	9	3

Question 1 [1 pt]. Insérer les clés dans la table. Donner un schéma de la table résultat. Indiquer les éventuelles collisions.

Question 2 [1 pt]. La fonction h_2 retourne un nombre impair dans tous les cas. Est-ce suffisant pour trouver un alvéole libre s'il en existe au moins un dans la table ?

Question 3 [2 pts]. On s'intéresse à l'algorithme de la recherche d'un élément x dans un tableau T de n éléments. Que serait un pire des cas ? un meilleur des cas ?

2 Problème (16 points)

On s'intéresse à la gestion des « facturettes » dans une grande surface. Une facturette est la donnée d'un numéro de client, d'un code produit et d'un montant. Le numéro de client et le code produit sont des entiers strictement positifs. Le montant est un flottant strictement positif.

Question 4 [2 pts]. Donner la déclaration d'un type `struct facturette`.

On souhaite enregistrer l'ensemble des facturettes émises sur une journée dans un dictionnaire représenté sous la forme d'un arbre binaire de recherche.

Question 5 [2 pts]. Donner la déclaration du type `struct abr` correspondant.

On souhaite gérer le dictionnaire de la façon suivante. On souhaite pouvoir retrouver une facturette à partir du couple (numéro de client, code produit). Il peut arriver qu'un même client achète plusieurs fois le même produit. Dans ce cas, on cumule les montants correspondant à ces achats sur une même facturette.

Enfin, on fait le souhait « technique » suivant : lors d'un parcours gauche-racine-droite du dictionnaire, les facturettes émises pour un même client doivent apparaître consécutivement.

Question 6 [2 pts]. Quelles sont les clés des facturettes ? Écrire une fonction `compare_facturette`, paramétrée par deux facturettes a et b, qui retourne -1, 0 ou 1 suivant que la clé de a est inférieure, égale ou supérieure à celle de b. Cette fonction, utilisée dans le cadre du dictionnaire, doit réaliser le souhait technique.

Question 7 [2 pts]. Donner le code d'une action ou d'une fonction qui ajoute une facturette a à un dictionnaire. Bien gérer le cas où le dictionnaire comporte déjà une facturette ayant même couple (numéro de client, code produit) que a.

Question 8 [2 pts]. Écrire une fonction `montant_client`, paramétrée par un dictionnaire et un numéro de client, qui retourne le montant total dépensé par le client (retourner zéro si le client est absent du dictionnaire). Tirer parti du souhait « technique » pour éviter de parcourir tout le dictionnaire.

2.1 Optimisation

Cette section est fortement inspirée de la technique des filtres de Bloom [1]. Après mise en application du dictionnaire, on s'est aperçu que, dans de très nombreux cas, la fonction de `montant_client` est appelée avec un client dont le numéro est absent du dictionnaire. Une séance de *brain storming* est organisée pour améliorer la structure de données, afin de réduire les temps de calculs.

Un collègue a l'idée suivante : associer au dictionnaire un tableau B de N booléens ainsi qu'une fonction f qui, à un numéro de client, associe un indice dans l'intervalle $[0, N-1]$. Initialement, les cases de B sont initialisées à *faux*. À chaque fois qu'une facturette (de numéro de client c) est enregistrée dans le dictionnaire, on calcule $i = f(c)$ et on met à *vrai* la case B_i. Lors des appels à `montant_client`, on pourra calculer $i = f(c)$ (où c désigne le numéro de client) et tester si B_i vaut *vrai* ou *faux*.

Un autre collègue fait remarquer qu'on peut choisir N nettement supérieur au nombre de clients mais que ça n'implique pas que les numéros de clients soient tous inférieurs à N : ce sont des numéros compliqués, pas consécutifs, qui nous sont imposés.

Question 9 [2 pts]. L'idée proposée peut effectivement être utilisée mais à condition de s'y prendre avec soin, sous peine d'obtenir des résultats faux. L'idée se rapproche d'une méthode étudiée en cours. Laquelle ? Quelle est la bonne façon de l'utiliser ?

Question 10 [2 pts]. Écrire une structure de données `struct DFB`, qui regroupe le dictionnaire des questions précédentes ainsi que le tableau de booléens.

Question 11 [2 pts]. Écrire une fonction `montant_client_DFB`, paramétrée par la structure de données précédente et un numéro de client, qui retourne le montant total dépensé par le client (retourner zéro si le client est absent dictionnaire). Vous pouvez supposer l'existence des fonctions `f` et `montant_client`. Votre solution doit utiliser le tableau de booléens.

Références

[1] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7) :422–426, 1970.