

1 Mécanismes de base des tables de hachage

On considère [1, Exercice 11.4.1, page 238] l'insertion des clés 10, 22, 31, 4, 15, 28, 17, 88, 59 dans une table de hachage de $N = 11$ alvéoles. La table est gérée avec la technique du double hachage. La fonction de hachage est $h(s) = (h_1(s), h_2(s)) = (s \bmod N, 1 + s \bmod (N - 1))$.

Question 1. Insérer les alvéoles dans la table.

Question 2. Il y avait 9 clés à insérer dans une table de 11 alvéoles. Était-on certain, pour autant, de trouver un alvéole libre pour chaque clé ?

Question 3. Pour plus de sûreté, un étudiant propose d'agrandir la table en prenant (mettons) $N = 24$ alvéoles. Qu'en pensez-vous ?

Question 4. Pour plus de rapidité, un autre étudiant propose de prendre $h_2(s) = s \bmod (N - 1)$ au lieu de $h_2(s) = 1 + s \bmod (N - 1)$. Qu'en pensez-vous ?

Question 5. Montrer que si $h_2(s)$ est premier avec N , alors, quel que soit l'indice $0 \leq x < N$, il existe un entier i tel que $h_1(s) + i h_2(s) = x \bmod N$. En déduire que, si $h_2(s)$ est premier avec N et si la table de hachage T comporte un alvéole libre, alors l'algorithme d'ajout dans T est certain de le trouver. On rappelle que,

- si $a = b \bmod N$ et $c = d \bmod N$ alors $a + c = b + d \bmod N$ et $a \times c = b \times d \bmod N$,
- si $h_2(s)$ et N sont premiers entre eux alors, il existe deux entiers u et v tels que $u h_2(s) + v N = 1$ (identité de Bézout).

2 Étude du nombre de comparaisons de chaînes de caractères

Dans cette section, on suppose que les éléments à enregistrer dans les tables de hachage sont des chaînes de caractères.

Question 6. On s'intéresse à une table de hachage T , d'environ 4000 alvéoles. Discuter les avantages et inconvénients des fonctions de hachage suivantes :

1. $h(s)$ = la somme des codes ASCII des caractères de s , modulo N .
2. $h(s)$ = la somme des codes ASCII des caractères de s fois 30, modulo N .
3. $h(s)$ = l'adresse de la chaîne s modulo N .

Proposer une autre fonction de hachage, qui vous semble meilleure.

On considère une table de hachage T disposant de N alvéoles, dont n sont occupés. Le taux de remplissage de la table est le rapport $\alpha = n/N < 1$. On suppose que les valeurs de hachage retournées par la fonction de hachage sont tirées de façon équiprobable dans l'intervalle $[0, N - 1]$. On s'intéresse au nombre de comparaisons de chaînes de caractères effectuées lors de la recherche d'une chaîne s (pour simplifier, on peut supposer qu'il n'appartient pas à T).

Question 7. Quelle est, en fonction de α , la probabilité que l'alvéole d'indice $h(s)$ soit vide ?

Question 8. Remplir (en fonction de α) le tableau suivant, où X désigne la variable aléatoire qui compte le nombre de comparaisons de chaînes de caractères effectuées lors de la recherche.

valeurs x_i de X	0	1	2	3	4
$p(X = x_i)$					

Question 9. En déduire que l'espérance $E(X) < \frac{\alpha}{1 - \alpha}$.

3 Redimensionnement d'une table de hachage

Quand une table de hachage devient trop pleine, il peut être utile de la redimensionner.

Question 10. Un étudiant propose de doubler la taille du tableau T et de recopier à l'identique le contenu de l'ancien tableau dans la première moitié du nouveau (grâce à `realloc`, par exemple). Qu'en pensez-vous ?

Question 11. Un autre étudiant propose de redimensionner le tableau T quand la fonction de hachage se comporte mal sur les données en entrée (nombre anormal de collisions). Qu'en pensez-vous ? Que pourrait être un « nombre anormal de collisions » ?

Question 12. Pour éviter de parcourir tout le tableau des alvéoles au moment du redimensionnement, une autre idée consiste à associer un deuxième tableau (appelons-le U) au tableau T . À chaque ajout d'un élément dans la table, on le range non seulement dans un alvéole de T mais on mémorise aussi son emplacement dans U . Les valeurs rangées dans U seraient rangées dans des emplacements consécutifs. En parcourant U , il serait alors facile d'accéder à toutes les chaînes présentes dans la table. Comment structureriez-vous tout cela ? Proposer des définitions de types. Expliquer le fonctionnement de la structure de données.

Références

- [1] Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein. *Introduction à l'algorithme*. Dunod, Paris, 2ème édition, 2002.