

## 1 Mécanismes de base des tables de hachage

On considère [?, 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 clés 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  $h_2(s)$  et  $N$  sont premiers entre eux alors, il existe deux entiers relatifs  $u$  et  $v$  tels que (identité de Bézout) :

$$u h_2(s) + v N = 1.$$

## 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.

## 3 Implantation en C

On considère les déclarations données en Figure 1.

```

#define N 11
struct valeur_hachage {
    int h1;
    int h2;
};
typedef struct valeur_hachage fonction_hachage (int);
enum etat_alveole { libre, occupe };
struct alveole {
    enum etat_alveole etat;
    int valeur;
};
struct table_hachage {
    struct alveole tab [N];
    fonction_hachage* h;
    int n;
};

```

FIGURE 1 – Implantation d’une table de hachage utilisant la technique du double hachage.

**Question 7.** Coder en C la fonction de hachage du début de la section 1.

**Question 8.** Écrire un extrait d’un programme principal qui déclare une table de hachage et l’initialise avec la fonction codée dans la question précédente.

**Question 9.** Écrire un extrait de la fonction `ajouter_table_hachage`.