

# Structures de données et algorithmes

## Répétition 8: Dictionnaires — Tables de hachage

Romain MORMONT

20 décembre 2019

### Exercice 1

Soit une table de hachage  $D$  initialement vide, d'une capacité de 11 éléments et dont les collisions sont gérées au moyen de listes simplement liées. Illustrer graphiquement l'insertion des clés

$\{23, 56, 44, 13, 88, 94, 16, 27, 9\}$

La fonction de hachage utilisée est  $h(k) = (2k + 5) \bmod 11$ .

### Exercice 2

Soit une table de hachage  $D$  initialement vide, d'une capacité de 11 éléments et dont les collisions sont gérées par sondage linéaire avec la fonction de hachage  $h(k, i) = (h'(k) + i) \bmod 11$  (la fonction de hachage  $h'$  est la même que dans l'exercice précédent). Illustrer graphiquement l'insertion des clés

$\{23, 56, 44, 13, 88, 94, 16, 27, 9\}$

.

### Exercice 3

Soit une table de hachage à adressage ouvert  $D$  initialement vide, d'une capacité initiale de 4 éléments avec un facteur de charge  $\alpha = 0.75$  (lorsque la charge dépasse 75%, on copie le contenu de la table dans une nouvelle de capacité double). La fonction de hachage utilisée est

$$h(k, i) = (h'(k) + i) \bmod m$$

où  $m$  représente la capacité courante de la table. Illustrer graphiquement l'insertion des clés

$\{23, 56, 44, 13, 88, 94, 16, 27, 9\}$

avec et sans rehachage lors de la copie.

### Exercice 4

Dans le JDK1.1, la fonction de hachage d'une chaîne de caractères était implémentée de la façon suivante. Quel est le défaut de cette fonction ?

```
public int hashCode() {
    int hash = 0;
    int skip = length() / 8;
    for (int i = 0; i < length(); i = i + skip) {
        hash = (hash * 37) + charAt(i);
    }
    return hash;
}
```

## Exercice 5

Désormais, la fonction de hachage d'une chaîne de caractères est implémentée en Java de la façon suivante :

```
public int hashCode() {
    int hash = 0;
    for (int i = 0; i < length(); i++) {
        hash = (hash * 31) + charAt(i);
    }
    return hash;
}
```

Trouvez des chaînes de caractères de longueur  $N$  produisant le même hash.