TD4 : tables de hachage

Exercice 1 : Adressage ouvert

On veut ranger dans une table de hachage des couples <clé,valeur> dont les clés sont les suivantes : 231, 45, 1529, 9, 67, 333, 51. La table est de taille 10 et on utilisera le chiffre des dizaine modulo la taille de la table pour fonction de hachage.

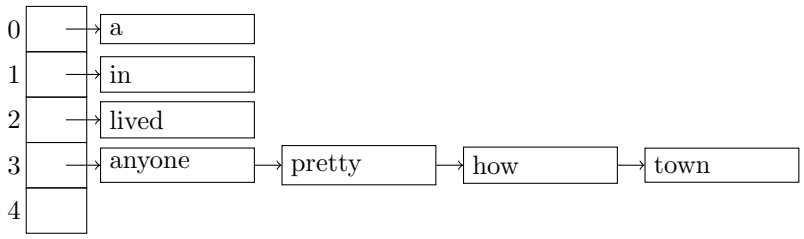
Q 1.1 Calculer les adresses associées aux clés que l'on souhaite insérer.

Q 1.2 Lister le parcours des alvéoles et compter le nombre de comparaisons faites pour l'insertion des éléments dans les cas où :

1. on utilise un hachage linéaire,
2. on utilise un hachage quadratique,
3. on utilise un hachage double (choisir soi-même la seconde fonction de hachage),

Exercice 2 : Résolution par chaînage

On suppose disposer d'une table de hachage de taille $M = 5$ où ne sont stockées que des clés dont le contenu est décrit sur le schéma ci-dessous.



Les clés sont des chaînes de caractères et on supposera qu'elles ont une longueur maximale : `type CLE is STRING[1..10].` Les clés sont complétées à gauche par des espaces si nécessaire. La fonction transformant la clé en un entier est obtenue en réalisant la somme des rangs de chaque caractère de la clé dans la table ASCII[1].

Q 2.1 Ecrire en Pascal la fonction de hachage calculant l'adresse de l'alvéole dans laquelle la clé doit être rangée (la fonction `function ord(c : CHAR) : CARDINAL` de Pascal fournit le numéro de caractère dans la table ASCII).

Q 2.2 Compléter la table ci-dessus avec les mots : with, up, so, floating, many, bells, down.

Q 2.3 Discuter comment vont se répartir les clés lors de l'insertion dans la table lorsque :

- $M = 10$ et $h(k)$ est le reste de la division de la longueur de k par 10,
- $M = 128$ et $h(k)$ est le résultat de la fonction `ord` appliquée au dernier caractère de k,
- $M = 10$ et $h(k)$ est le reste de la division par 10 de la somme des rangs des caractères élevée au carré.

[1] 'a' est en position 97 et 'z' en position '122', ' ' est en position 32

Exercice 3 : Une utilisation des tables de hachages pour trier

Dans cet exercice, nous allons utiliser des tables de hachage à résolution par chaînage afin d'implanter un algorithme de tri. Contrairement au cours, on insèrera en queue de liste et non en tête de liste.

Nous cherchons à trier des couples `MATRICULES` dans l'ordre croissant des matricules. Rappelons que les `MATRICULES` sont des tableaux de chiffres de longueur fixe M.

L'algorithme proposé est le suivant :

```
Soit D un tableau contenant tous les matricules
Pour i allant de 1 à M faire
  Soit T une table de hachage de taille 10 dont la
    fonction de hachage est h(k) = le i-ème chiffre de k
  Pour chaque matricule k>
     Ajouter le couple k à T
  Fin Pour
  Vider D
  Ranger tous les couples de T dans le tableau D en parcourant les
    alvéoles de 0 à 9
Fin Pour
Imprimer le contenu de d
```

Q 3.1 Tester l'algorithme sur les données suivantes : 231, 45, 529, 9, 67, 333, 51, 21. Pour chaque étape i, dessiner la table T et le tableau D.

Q 3.2 Pourquoi est-il important d'ajouter en queue dans les listes ?

Q 3.3 Traduire en Pascal l'algorithme donné ci-dessus (on utilisera les fonctions sur les listes vues en API2)

Q 3.4 Quelle est la complexité en espace de cet algorithme ?

Q 3.5 Quelle est la complexité en temps de cet algorithme ?

Q 3.6 Quels changements faudrait-il opérer pour trier en ordre décroissant ?