

Algorithmique

Correction Contrôle n° 1 (C1)

INFO-SPÉ (S3) – EPITA

5 Nov. 2013 - 10 :00

Solution 1 (Hachage linéaire – 2 points)

Représentation des structures de données dans le cas du hachage linéaire (*Hachage linéaire avec un coefficient de décalage $d = 4$*) voir tableau 1 :

FIGURE 1 – Hachage linéaire

0	sisko
1	odo
2	quark
3	neelix
4	data
5	kirk
6	q
7	picard
8	worf
9	tuvok
10	

Solution 2 (Hachage : Questions... – 3 points)

1. Les trois propriétés essentielles que doit posséder une fonction de hachage sont :
 - (a) Uniforme
 - (b) Déterministe
 - (c) Facile et rapide à calculer
2. Une collision secondaire est due au fait que deux éléments se retrouvent en collision sur une case du tableau de hachage alors que leur valeur de hachage primaire est différente (cf. Hachage coalescent).
3. Le phénomène provoqué par le hachage linéaire est le regroupement ou accumulation d'éléments (clustering) que l'on peut résoudre en envisageant un double hachage.

Solution 3 (Hachage : Tableaux valides – 3 points)

Les tableaux ne pouvant pas résulter d'une insertion quelconque des clés sont : A-B-D
Le seul valable est le C qui pourrait correspondre à la séquence d'ajout {B, E, A, C, D, F, G} (*il y en a d'autres*).

Solution 4 (Des croissants – 5 points)

Spécifications :

print234 (t_a234 A) affiche en ordre décroissant les clés de l'arbre A non vide.

```
algorithme procedure print234
  parametres locaux
    t_a234    A

  variables
    entier i

  debut
    si A↑.fils[1] = NUL alors
      pour i ← A↑.nbcles décroissant jusqu'à 1 faire
        écrire (A↑.cle[i], ' ')
      fin pour
    sinon
      pour i ← A↑.nbcles décroissant jusqu'à 1 faire
        print234 (A↑.fils[i+1])
        écrire (A↑.cle[i], ' ')
      fin pour
      print234 (A↑.fils[1])
    fin si
  fin algorithme procedure print234
```

Solution 5 (Arbre 2-3-4 : insertion – 7 points)

Spécifications :

La fonction `insert_234 (x, A)` insère la clé x dans l'arbre A de type `t_a234`, sauf si celle-ci est déjà présente. Elle retourne un booléen indiquant si l'insertion a eu lieu. L'arbre A n'est pas vide, et sa racine n'est pas un 4-nœud.

```
algorithme fonction insert_234 : booléen
  parametres locaux
    t_element    x
    t_a234       A

  variables
    entier       i, j

  debut

    i ← 1
    tant que (i ≤ A↑.nbcles) et (x > A↑.cle[i]) faire
      i ← i+1
    fin tant que

    si (i ≤ A↑.nbcles) et (x = A↑.cle[i]) alors
      retourne faux
    sinon
      si A↑.fils[1] = NUL alors

        pour j ← A↑.nbcles décroissant jusqu'à i faire
          A↑.cle[j+1] ← A↑.cle[j]
        fin pour
        A↑.nbcles ← A↑.nbcles + 1
        A↑.fils[A↑.nbcles+1] ← NUL
        A↑.cle[i] ← x
        retourne vrai

      sinon
        si A↑.fils[i]↑.nbcles = 3 alors
          si A↑.fils[i]↑.cle[2] = x alors
            retourne faux
          fin si
          eclate (A, i)
          si x > A↑.cle[i] alors
            i ← i + 1
          fin si
        fin si

      retourne insert_234 (x, A↑.fils[i])
    fin si
  fin si
fin algorithme fonction insert_234
```

