utfcode
utf8.sty 3.10 UTF-8 input encoding 13.06.2000
scanner for code UTF-8 installed.

Techniques de hachage : TD Semaines 4–5

Version du 15 octobre 2012

1 Hachage dynamique

Exercice 1 – Quelques exemples

Dans cet exercice, les Ã©lÃ©ments sont les lettres de l'alphabet : $a, b, \dots y, z$.
Leurs valeurs de hachage sont $h(a) = 00000, h(b) = 00001, \dots, h(y) = 11000, h(z) = 11001$.

Lettre	Clef	Lettre	Clef	Lettre	Clef	Lettre	Clef
a	00000	i	01000	q	10000	y	11000
b	00001	j	01001	r	10001	z	11001
c	00010	k	01010	s	10010		
d	00011	l	01011	t	10011		
e	00100	m	01100	u	10100		
f	00101	n	01101	v	10101		
g	00110	o	01110	w	10110		
h	00111	p	01111	x	10111		

1. RÃ©aliser le hachage dynamique des clefs $t, m, y, u, n, r, p, x, e, s, i, b$, dans cet ordre, avec pages de taille 4. Que se passe-t-il si on modifie l'ordre d'insertion des clefs ?
2. RÃ©aliser le hachage dynamique des clefs $a, b, c, d, e, f, g, h, i, j, k, l, m$, dans cet ordre, avec pages de taille 4, puis avec pages de taille 7.

Exercice 2 – Recherche et insertion

Pour travailler sur le hachage dynamique, on utilisera les primitives suivantes (vous en complÃ©terez la spÃ©cification) :

`IndexArbre`	*index* $\times$ *index* $\to$ *index*
`IndexFeuille`	*page* $\to$ *index*
`PageVide`	$\to$ *page*
`IndexGauche`	*index* $\to$ *index*
`IndexDroit`	*index* $\to$ *index*
`PageDeFeuille`	*index* $\to$ *page*
`ÃlÃ©ment`	*page* $\times$ *entier* $\to$ *Ã©lÃ©ment*
`EstFeuille`	*index* $\to$ *boolÃ©en*
`EstPagePleine`	*page* $\to$ *boolÃ©en*
`EstDansPage`	*Ã©lÃ©ment* $\times$ *page* $\to$ *boolÃ©en*
`InsertionDansPage`	*page* $\times$ *Ã©lÃ©ment* $\to$ *page*

On suppose que l'on dispose aussi d'une fonction :

`BitHachage`	*Ã©lÃ©ment* $\times$ *entier* $\to$ *bit*
`BitHachage(x,k)`	*renvoie le k-iÃ¨me bit de la valeur de hachage de x*

1. (Recherche) Ãcrire un algorithme de recherche dans un index.
2. (Insertion) Ãcrire un algorithme d'insertion dans un index.

2 Familles de fonctions de hachage

Exercice 3 – Hachage k-universel

Soit $\mathcal{H}$ une famille de fonctions de hachage dans laquelle chaque fonction $h \in \mathcal{H}$ envoie l'univers de clefs U dans l'intervalle d'entiers $[0, 1, \dots, m-1]$. On dit que $\mathcal{H}$ est **k-universelle** ssi, pour toutes clefs $x_1, \dots, x_k$ deux ï¿½ deux distinctes et pour toutes valeurs $v_1, \dots, v_k$ dans $[0, 1, \dots, m-1]$:

$$\mid \{h \in \mathcal{H}; h(x_1) = v_1, \dots, h(x_k) = v_k\} \mid = \frac{\mid \mathcal{H} \mid}{m^k}.$$

Autrement dit, en munissant $\mathcal{H}$ de la probabilitÃ© uniforme :

$$\Pr(\{h \in \mathcal{H}; h(x_1) = v_1, \dots, h(x_k) = v_k\}) = \frac{1}{m^k}$$

ou encore, pour h choisie au hasard dans $\mathcal{H}$:

$$\Pr(h(x_1) = v_1, \dots, h(x_k) = v_k) = \frac{1}{m^k}.$$

1. Soit U un univers ayant n clefs et $\mathcal{H}$ la famille de toutes les fonctions de U dans $[0, 1, \dots, m-1]$. Montrer que $\mathcal{H}$ est k-universelle (pour $k \leqslant n$).
2.
   1. Écrire la dÃ©finition de famille 2-universelle.
   2. Montrer que si une famille $\mathcal{H}$ de fonctions de hachage est 2-universelle alors elle vÃ©rifie, pour toute clef x et pour toute valeur v dans $[0, 1, \dots, m-1]$: $\Pr(h(x) = v) = \frac{1}{m}$.
   3. Montrer que si une famille $\mathcal{H}$ de fonctions de hachage est 2-universelle alors elle est universelle.
   4. Soit $U = \{x_1, x_2, x_3, x_4\}$, on considÃ¨re les familles $\mathcal{H}_0$ et $\mathcal{H}_1$ de fonctions de U dans $\{0, 1\}$ donnÃ©es par les tableaux suivants :

$\mathcal{H}_0$	h_1	h_2	h_3	h_4
x_1	0	0	0	0
x_2	0	1	0	1
x_3	0	0	1	1
x_4	0	1	1	0

$\mathcal{H}_1$	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8
x_1	0	1	0	1	0	1	0	1
x_2	0	1	1	0	0	1	1	0
x_3	0	1	0	1	1	0	1	0
x_4	0	1	1	0	1	0	0	1

   La famille $\mathcal{H}_0$ est-elle universelle ? 1-universelle ? 2-universelle ? et la famille $\mathcal{H}_1$?

   5. Une famille universelle est-elle nÃ©cessairement 2-universelle ?
   6. Montrer que si une famille $\mathcal{H}$ de fonctions de hachage est $(k+1)$-universelle alors elle est k-universelle.
   7. En dÃ©duire que si une famille $\mathcal{H}$ de fonctions de hachage est k-universelle, avec $k \geqslant 2$, alors elle est universelle et elle vÃ©rifie, pour toute clef x et pour toute valeur v dans $[0, 1, \dots, m-1]$: $\Pr(h(x) = v) = \frac{1}{m}$.
3. Supposons que l'univers U est l'ensemble des n-uplets de valeurs de $\mathbb{Z}_p = \{0, 1, \dots, p-1\}$, où p est premier ; autrement dit $U = (\mathbb{Z}_p)^n$. Soit $x = \langle x_0, \dots, x_{n-1}\rangle \in U$. Pour tout n-uplet $a = \langle a_0, \dots, a_{n-1}\rangle \in U$, définissons la fonction de hachage $h_a : U \to \mathbb{Z}_p$ par :

$$h_a(x) = \left(\sum_{j=0}^{n-1} a_j x_j\right) \bmod p.$$

   Soit $\mathcal{H} = \{h_a \ ; \ a \in U\}$.

   1. Calculer h_a pour $p = 2$, $n = 3$, $a = \langle 1, 0, 1\rangle$.
   2. Montrer que $(a \neq b) \Rightarrow (h_a \neq h_b)$. En dÃ©duire que $\mid \mathcal{H} \mid = p^n$. La famille $\mathcal{H}$ est-elle Ã©gale ï¿½ l'ensemble de toutes les fonctions de U dans $\mathbb{Z}_p$?

3. Montrer que $\mathcal{H}$ est universelle mais pas 2-universelle. (Indication : trouver une clef pour laquelle toutes les fonctions de hachage de $\mathcal{H}$ produisent la même valeur.)

4. On modifie légèrement $\mathcal{H}$ par rapport à la question précédente :
pour tout n-uplet $a = \langle a_0, \ldots, a_{n-1} \rangle \in U$ et $b \in \mathbb{Z}_p$, on définit la fonction de hachage $h_{a,b}$ par :

$$h_{a,b}(x) = \left(\sum_{j=0}^{n-1} a_j x_j + b \right) \bmod p.$$

Soit maintenant $\mathcal{H}' = \{h_{a,b}\ ;\ a \in U,\ b \in \mathbb{Z}_p\}$.

1. Calculer $h_{a,b}$ pour $p = 2$, $n = 3$, $a = \langle 1, 0, 1 \rangle$, $b = 1$.
2. Montrer que $(h_{a,b} = h_{a',b'}) \Rightarrow (a = a'$ et $b = b')$. En dÃ©duire que $|\mathcal{H}'| = p^{n+1}$.
3. Montrer que $\mathcal{H}'$ est 2-universelle.

5. (Application Cryptographie) Soit $\mathcal{H}$ une famille telle que chaque fonction de hachage de $\mathcal{H}$ envoie l'univers des clefs U sur $\mathbb{Z}_p$ où p est premier. Supposons qu'Alice et Bob s'entendent secrètement sur une fonction de hachage $h \in \mathcal{H}$. Alice envoie ensuite un message $m \in U$ à Bob, à travers l'internet ; elle authentifie ce message en envoyant aussi le tag t, $t = h(m)$. Pour vérifier que le message vient bien d'Alice, Bob vérifie qu'il a bien $h(m_{\text{reÃ§u}}) = t_{\text{reÃ§u}}$. Supposons qu'un intrus essaye de tromper Bob en interceptant la paire (m, t) et en la remplaÃ§ant par une autre paire (m', t'). On veut montrer que la probabilitÃ© qu'il rÃ©ussisse est inférieure ou Ã©gale à $\frac{1}{p}$ lorsque $\mathcal{H}$ est 2-universelle.

1. Montrer que si $\mathcal{H}$ est la famille de fonctions h_a dÃ©finie dans la question , alors dans certains cas l'intrus est sÃ»r de faire accepter ï¿½ Bob la paire (m', t').
2. On considÃ¨re la famille $\mathcal{H}$ donnÃ©e par le tableau suivant ($p = 2$) :

$\mathcal{H}$	h_1	h_2	h_3	h_4
x_1	1	1	0	0
x_2	0	1	0	1
x_3	1	0	0	1
x_4	1	0	1	0

VÃ©rifier que $\mathcal{H}$ est 1-universelle. Existe-t-il des configurations oÃ¹$l'intrustrompeBobavecuneprobabilitÃ©supÃ$

3. Démontrer que si $\mathcal{H}$ est 2-universelle alors la probabilité que Bob soit trompÃ© est toujours inférieure ou Ã©gale à $\frac{1}{p}$ (et ce même s'il connaît la famille $\mathcal{H}$, et quelle que soit sa capacité de calcul).

Exercice 4 – Hachage universel

Soit p un nombre premier, on considÃ¨re l'univers des clefs $U = \{0, \ldots, p-1\}$. Soit m un entier tel que $p > m$. Pour $a \in \mathbb{Z}_p^*$ et $b \in \mathbb{Z}_p$, on dÃ©finit $f_{a,b} : U \to \{0, \ldots, m-1\}$ par

$$f_{a,b}(x) = ((ax + b) \bmod p) \bmod m.$$

On munit $\mathbb{Z}_p^* \times \mathbb{Z}_p$ de la probabilitÃ© uniforme.

1. Montrer que, pour toutes clefs distinctes x et y de U :

$$\Pr(\{(a,b) \in \mathbb{Z}_p^* \times \mathbb{Z}_p\ ;\ f_{a,b}(x) = f_{a,b}(y)\}) \leqslant \frac{1}{m}.$$

Indication
Mettre en bijection $\{(a,b) \in \mathbb{Z}_p^* \times \mathbb{Z}_p\ ;\ f_{a,b}(x) = f_{a,b}(y)\}$ et $\{(r,s) \in \mathbb{Z}_p \times \mathbb{Z}_p\ ;\ r \neq s$ et $r = s \bmod m\}$.

3 Le hachage coucou

Exercice 5 – PrÃ©sentation

Le *hachage coucou* est une technique de hachage qui utilise deux fonctions de hachage h_1 et h_2. Ces deux fonctions sont dÃ©finies sur un univers de n clefs et sont ï¿½ valeurs dans $\{1, \ldots, r\}$, avec $r > n$. On suppose que :
- h_1 et h_2 rÃ©partissent uniformÃ©ment les clefs, *i.e.* $\Pr(h_1 = i) = \Pr(h_2 = i) = \frac{1}{r}$ pour tout $i \in \{1, \ldots, r\}$;
- h_1 et h_2 sont indÃ©pendantes, *i.e.* $\Pr(h_1 = i, h_2 = j) = \Pr(h_1 = i)\Pr(h_2 = j)$ pour tous $i, j \in \{1, \ldots, r\}$.

Les clefs sont rÃ©parties dans une table de hachage $T[1..r]$, chaque clef x peut Ãªtre placÃ©e ï¿½ la position $h_1(x)$ ou ï¿½ la position $h_2(x)$ dans la table T. Pour insÃ©rer une clef x dans la table T, on calcule sa position $i = h_1(x)$. Si la case $T[i]$ est vide on y met la clef x, sinon on Ã©jecte la clef y dÃ©jï¿½ prÃ©sente et on la remplace par x. Il faut maintenant placer la clef y dans la table. Pour cela on calcule l'autre position j de y (si $i = h_1(y)$ alors $j = h_2(y)$ sinon $j = h_1(y)$). On recommence avec y ce qu'on a fait avec x (si $T[j]$ est vide on y met y, sinon...). Le processus s'arrÃªte lorsqu'on tombe sur une case vide ou lorsqu'on a atteint le nombre maximal d'itÃ©rations, que l'on a fixÃ© au prÃ©alable (Ã©gal au nombre n de clefs). Dans ce dernier cas deux nouvelles fonctions de hachage sont choisies et on reconstruit toute la table (on dit qu'il y a un *re-hachage*). Il est possible que ce re-hachage n'aboutisse pas lui non plus, auquel cas on a recours ï¿½ un deuxiÃ¨me re-hachage (et Ã©ventuellement ï¿½ un troisiÃ¨me, etc).
Voici un pseudo-code pour la procÃ©dure d'insertion d'une clef x dans une table T.

```
InsÃ©rer(T, x)
    Si T[h1(x)] <> x Et T[h2(x)] <> x Alors
        pos <- h1(x)
        RÃ©pÃ©ter n fois
            Si T[pos] est vide Alors
                T[pos] <- x
                Exit InsÃ©rer
            Sinon
                Ã©changer x et T[pos]
                Si pos = h1(x) Alors pos <- h2(x) Sinon pos <- h1(x) Fin Si
            Fin Si
        Fin RÃ©pÃ©ter
        Re-hacher(T)
        InsÃ©rer(T,x)
Fin InsÃ©rer
```

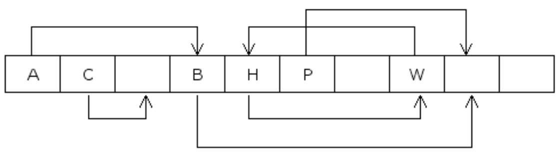

FIGURE 1 – Hachage coucou

1. La figure ?? reprÃ©sente le hachage coucou des clefs A, B, C, H, P, W dans une table $T[1..10]$. Chaque clef x est placÃ©e ï¿½ une position correspondant ï¿½ l'une de ses deux valeurs de hachage et une flÃ¨che indique l'autre position possible de x dans la table (correspondant ï¿½ l'autre valeur de hachage de x).
   (a) RÃ©aliser l'insertion de la clef Z ayant comme valeurs de hachage $h_1(Z) = 5$ et $h_2(Z) = 1$.
   (b) Peut-on insÃ©rer une clef V ayant comme valeurs de hachage $h_1(V) = 5$ et $h_2(V) = 8$?
2. Si x est un entier, on dÃ©signe par $b_0(x), b_1(x), \ldots, b_k(x)$ les bits de x dans l'Ã©criture binaire de x. Autrement dit, $x = b_k(x)2^k + \ldots b_1(x)2^1 + b_0(x)2^0$, avec $b_i(x) = 0$ ou 1. On veut rÃ©aliser le hachage coucou de clefs entiÃ¨res en utilisant les opÃ©rations & et `rot` ainsi dÃ©finies :
   - si x et y sont deux entiers naturels alors $x \& y$ est l'entier z tel que $b_i(z) = 1$ ssi $b_i(x) = 1$ et $b_i(y) = 1$.

– si x est un entier naturel alors $\texttt{rot}(x, j)$ est l'entier obtenu en faisant une rotation circulaire de j bits vers la droite dans la reprÃ©sentation binaire de x. Autrement dit, si $x = a_k 2^k + \dots a_1 2^1 + a_0 2^0$ et si $z = \texttt{rot}(x, j)$, avec $j \leqslant k$, alors $z = a_{j-1} 2^k + \dots a_0 2^{k-j+1} + a_k 2^{k-j} + \dots + a_j 2^0$.

(a) Calculer 13 & 23 et $\texttt{rot}(100, 4)$.

(b) On considÃ¨re les deux fonctions de hachage suivantes, ï¿½ valeurs dans $\{1, \dots, 8\}$:
– $h_1(x) = [(x^2 \bmod 17) \,\&\, 7] + 1$
– $h_2(x) = [(\texttt{rot}(x, 4) \bmod 33) \,\&\, 7] + 1$
Effectuer le hachage coucou des clefs 14, 100, 1000, 31, 117, dans cet ordre.
Aide : $14^2 \bmod 17 = 9$, $100^2 \bmod 17 = 4$,$1000^2 \bmod 17 = 9$, $31^2 \bmod 17 = 9$, $117^2 \bmod 17 = 4$.

Exercice 6 – Hachage coucou et graphe coucou

Dans cet exercice, on considÃ¨re que n clefs ont Ã©tÃ© rÃ©parties dans une table $T[1..r]$ par la mÃ©thode du hachage coucou, en utilisant deux fonctions de hachage h_1 et h_2.
Le *graphe coucou* est le graphe orientÃ© de sommets $1, \dots, r$ dans lequel il y a un arc de i vers j ssi il existe une clef x telle que $T[i] = x$ et j est l'autre valeur de hachage possible de x.
Remarques :
– s'il y a un arc de i vers j alors il existe une clef x telle que i et j sont les deux positions possibles de x, autrement dit une clef x telle que ($h_1(x) = i$ et $h_2(x) = j$) ou ($h_1(x) = j$ et $h_2(x) = i$) ;
– la rÃ©ciproque n'est pas vraie. Si l'on considÃ¨re l'exemple 1 de l'exercice 1, les deux positions possibles de la clef A sont 1 et 4 mais il n'y a pas d'arc de 4 vers 1 dans le graphe coucou.

Le graphe coucou a deux propriÃ©tÃ©s intÃ©ressantes :
– si l'insertion de la clef x est susceptible d'Ã©jecter la clef y alors il y a un chemin dans le graphe coucou de l'une des positions possibles de x vers la position de y ;
– si l'insertion d'une clef provoque un re-hachage alors il y a un circuit dans le graphe coucou.

On rappelle qu'un *circuit* est un chemin d'un sommet vers lui-mÃªme.

1. Dessiner le graphe coucou des exemples de l'exercice 1. Donner des exemples de circuit dans chacun de ces graphes.
2. Ãtant donnÃ©e une clef x, calculer la probabilitÃ© $P_{i,j}$ qu'elle soit placÃ©e ï¿½ la position i ou ï¿½ la position j, autrement dit la probabilitÃ© que ($h_1(x) = i$ et $h_2(x) = j$) ou ($h_1(x) = j$ et $h_2(x) = i$). Ne pas oublier le cas oÃ¹i=j.

Ãtant donnÃ©es n clefs, majorer la probabilitÃ© que l'une d'entre elles soit placÃ©e ï¿½ la position i ou ï¿½ la position j.
En dÃ©duire une majoration de la probabilitÃ© d'avoir un arc de i vers j dans le graphe coucou.

Montrer le lemme suivant :
– pour tout entier $l \geqslant 1$, pour toutes positions i et j et pour toute constante $c > 1$, si $r > 2cn$ alors la probabilitÃ© que le plus court chemin de i vers j dans le graphe coucou soit exactement de longueur l est majorÃ©e par c^{-l}/r.

Indication : faire un raisonnement par rÃ©currence sur l.

On veut maintenant rÃ©aliser l'insertion de εn clefs dans la table (qui contient n clefs). On suppose que c est une constante telle que la taille r de la table vÃ©rifie $r > 2c(1 + \varepsilon)n$.

(a) Montrer que la probabilitÃ© d'avoir un circuit dans le graphe coucou lors de l'insertion des εn clefs est majorÃ©e par $\frac{1}{c-1}$. Indication : utiliser le lemme de la question **??**.

(b) Si $c = 3$, quelle est la probabilitÃ© de devoir faire un re-hachage ? puis un deuxiÃ¨me (dans le cas oÃ¹$le premier re-hachage aurait crÃ©Ã© un circuit) ? puis un troisiÃme ? puis un$ k$-iÃme ? En dÃ©duire le nombre moyen de re - ha$

Première partie

E

n admettant que le coÃ»t d'un re-hachage est en $O(n)$, montrer que le coÃ»t amorti moyen d'un re-hachage dans la suite des εn insertions est constant. On supposera que $c = 3$.