Algorithmique Avancée

TD 4 : Techniques de hachage

Rappels élémentaires de probabilités.

Il s'agit ici de donner trois exemples simples d'usage de probabilités pour donner certains rappels de notions abordées traditionnellement au lycée. L'exemple sur le hachage est repris dans le quatrième cours d'Algorithmique Avancée.

Définitions. On étudie une *expérience aléatoire* (lancer de dés, calcul d'une valeur de hachage, etc.). L'ensemble de toutes les issues possibles de cette expérience est appelé *univers* (ou *univers des possibles*). On note généralement cet ensemble Ω.

Rappel. Le *cardinal* de l'ensemble E est le nombre d'éléments contenus dans E, et est noté $|E|$ ou $\text{card}(E)$.

- Pour l'expérience du lancer d'une pièce, l'univers des possibles est $\Omega = \{\text{Pile}, \text{Face}\}$ et $|\Omega| = 2$.
- Pour l'expérience du lancer de *deux* dés (distinguables), l'univers des possibles est

$$\Omega = \{(1,1)\,;(1,2)\,;\ldots\,;(1,6)\,;(2,1)\,;\ldots\,;(5,6)\,;(6,6)\}\,,$$

  c'est à dire les paires ordonnées où la première composante est la face du premier dé, et la deuxième composante la face du deuxième dé. On a donc $|\Omega| = 6 \times 6 = 36$.
- Pour l'expérience du hachage d'une valeur vers l'intervalle entier $[0, m]$, l'univers des possibles est cet intervalle (quand on applique la fonction de hachage à une valeur, le résultat peut être n'importe laquelle des $m+1$ valeurs de l'intervalle $[0, m]$). On a donc $|\Omega| = |[0, m]| = m + 1$.

Définition. On appelle E un *événement*, si E est un sous-ensemble de Ω, ce qu'on note $E \subset \Omega$.

Proposition. La *probabilité* d'un élément E dans un univers Ω est donnée par,

$$\mathbb{P}[E] = \frac{|E|}{|\Omega|}.$$

- Dans le cas de la pièce,
  - on peut considérer l'événement E_1 « la pièce est tombée sur pile », c'est-à-dire $E_1 = \{\text{Pile}\}$, et alors $\mathbb{P}[E_1] = \frac{E_1}{\Omega} = \frac{1}{2}$ (car comme dit ci-dessus, pour l'expérience du lancé de pièce on a $\Omega = \{\text{Pile}, \text{Face}\}$) ;
  - on peut considérer l'événement E_2 « la pièce n'est tombée ni sur pile, ni sur face » (c'est-à-dire elle est tombée sur la tranche), et alors $E_2 = \{\}$ et $\mathbb{P}[E_2] = 0$.
- Dans le cas du lancer de dés,
  - on peut considérer l'événement E_3 « les dés ont la même valeur », c'est-à-dire

$$E_3 = \{(1,1)\,;(2,2)\,;(3,3)\,;(4,4)\,;(5,5)\,;(6,6)\}$$

    d'où $\mathbb{P}[E_3] = 6/36 = 1/6$;
  - on peut considérer l'événement E_4 « la somme des dés est inférieure ou égale à 3 », ainsi $E_4 = \{(1,1)\,;(1,2)\,;(2,1)\}$, d'où $\mathbb{P}[E_4] = 3/36 = 1/12$.
- Dans le cas du hachage de la valeur x par la fonction de hachage h, on peut se demander quand cette valeur, $h(x)$, est égale à m, c'est-à-dire l'événement $h(x) = m$, qu'on peut noter E_5 où $E_5 = \{m\}$ (un singleton ne contenant que m), d'où

$$\mathbb{P}[h(x) = m] = \mathbb{P}[E_5] = \frac{|E_5|}{|\Omega|} = \frac{1}{m+1}$$

  car on rappelle que pour l'expérience du hachage d'une valeur, $\Omega = [0, m]$. En fait, on peut voir que $\mathbb{P}[h(x) = k] = \frac{1}{m+1}$ pour toute valeur de hachage de l'intervalle d'entiers $[0, m]$.

Notion d'uniformité

Soit E un ensemble de possibilités, de cardinal $|E| = m$. On dit que le tirage d'un élément de E est *uniforme* si chaque élément de E à la même probabilité d'être tirée, autrement dit, si la probabilité de tirer un élément $x \in E$, est $1/m$.

1 Fonctions de hachage

Exercice 1.1 ***: Chaînes de caractères***

Dans cet exercice, les clés sont des chaînes composées de caractères minuscules non accentués.
Chaque caractère, de 'a' à 'z' est codé sur 5 bits :

caractère	codage	caractère	codage	caractère	codage	caractère	codage
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		

Question 1.1.1 On considère la fonction de hachage h ainsi définie : si $s = x_1x_2 \dots x_n$ et si c_i est le codage de x_i alors $h(s) = c_1$ **xor** c_2 **xor** $\dots$ **xor** c_n
Calculer les valeurs de hachage de 'et', 'ou', 'ni', 'car', 'arc' et les exprimer en base 10. Que peut-on dire lorsque deux mots sont anagrammes l'un de l'autre ?

Question 1.1.2 Pour éviter l'inconvénient rencontré avec les anagrammes, on décide de procéder ainsi :
— on décale les bits de c_1 circulairement de 1 rang vers la droite,
— on décale les bits de c_2 circulairement de 2 rang s vers la droite,
— on décale les bits de c_3 circulairement de 3 rangs vers la droite,
...
Puis on applique le **xor** aux décalages.
Calculer les valeurs de hachage de 'car', 'arc', 'lire', 'lier'.

Exercice 1.2 ***: Division***

Question 1.2.1 On considère la fonction de hachage h définie sur des entiers par $h(x) = x \mod m$. Pour $m = 37$, calculer $h(180)$, $h(501)$.

Question 1.2.2 Avec ce genre de fonctions, que se passe-t-il si m est pair et si toutes les clés sont paires ?

Exercice 1.3 ***: Multiplication***

On considère la fonction de hachage h définie sur des réels par $h(x) = \lfloor((x * \theta) \mod 1) * m\rfloor$. Pour $\theta = 0,6$ et $m = 30$, calculer $h(180)$, $h(501)$.

2 Résolution des collisions. Méthodes indirectes

Exercice 2.1 ***: Chaînage séparé***

Question 2.1.1 Dans cette question $m = 9$, donner le résultat du hachage par adjonctions successives des clés $e_1, e_2, \ldots, e_{13}$ ayant pour valeurs de hachage respectives $2, 0, 3, 0, 3, 0, 4, 8, 1, 5, 4, 2, 4$, en utilisant la méthode du hachage séparé.

Question 2.1.2 On hache n clés dans une table de taille m.
Quel est le coût d'une recherche négative dans le pire cas ? dans le meilleur cas ?
Quel est le coût d'une recherche positive dans le pire cas ? dans le meilleur cas ?
Comparer avec les résultats en moyenne vus en cours (on suppose la répartition uniforme).

3 Hachage dynamique

Exercice 3.2 ***: Quelques exemples***
Dans cet exercice, les éléments sont les lettres de l'alphabet. Leurs valeurs de hachage, ou clés sont indiquées dans ce tableau.

Lettre	Clé	Lettre	Clé	Lettre	Clé	Lettre	Clé
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		

Question 3.2.1 Réaliser le hachage dynamique des clés $t, m, y, u, n, r, p, x, e, s, i, b$, dans cet ordre, avec pages de taille 4.
Que se passe-t-il si on modifie l'ordre d'insertion des clés ?

Question 3.2.2 Réaliser le hachage dynamique des clés $a, b, c, d, e, f, g, h, i, j, k, l, m$, dans cet ordre, avec pages de taille 4, puis avec pages de taille 7.

Exercice 3.3 ***: Recherche et insertion***
Pour travailler sur le hachage dynamique, on utilisera les primitives suivantes (vous en compléterez la spécification) :

`IndexArbre`	*index* × *index* → *index*
`IndexFeuille`	*page* → *index*
`PageVide`	→ *page*
`IndexGauche`	*index* → *index*
`IndexDroit`	*index* → *index*
`PageDeFeuille`	*index* → *page*

`Élément`	*page* × *entier* → *élément*
`EstFeuille`	*index* → *booléen*
`EstPagePleine`	*page* → *booléen*
`EstDansPage`	*élément* × *page* → *booléen*
`InsertionDansPage`	*page* × *élément* → *page*

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

`BitHachage`	*élément* × *entier* → *bit*
`BitHachage(x,k)`	*renvoie le k-ième bit de la valeur de hachage de x*

Question 3.3.1 (Recherche) Écrire un algorithme de recherche dans un index.

Question 3.3.2 (Insertion) Écrire un algorithme d'insertion dans un index.

4 Familles de fonctions de hachage

Exercice 4.1 ***: Application des familles universelles***

Cet exercice étudie une stratégie de hachage qui, étant donné un ensemble *statique* de n clés, effectue une recherche avec une *complexité au pire* en $O(1)$ comparaisons entre clés, en utilisant une mémoire totale en $O(n)$ (la mémoire est comptée en nombre de cases pouvant contenir une clé).

On suppose que l'on dispose d'un ensemble universel de fonctions de hachage $\mathcal{H}$.

Question 4.1.1 Montrer que si l'on hache un ensemble statique de n clés dans une table de hachage de taille $m = n^2$, à l'aide d'une fonction de hachage h choisie aléatoirement dans un ensemble universel de fonctions de hachage, alors la probabilité qu'il n'y ait aucune collision est supérieure à $1/2$.
On pourra utiliser l'inégalité de Markov : $\mathbb{P}(X \geq a) \leq \mathbb{E}[X]/a$.

Question 4.1.2 On suppose que l'on peut réserver une table de hachage dont la taille est le carré du nombre d'éléments à stocker. Un algorithme permettant de hacher les éléments sans aucune collision (qui a donc une complexité en $O(1)$ dans le pire des cas) est obtenu en essayant plusieurs fonctions de hachage de $\mathcal{H}$. Montrer que le nombre moyen d'essais de fonctions est inférieur à 2.

Question 4.1.3 Lorsque l'on ne peut pas réserver une table de hachage dont la taille est le carré du nombre d'éléments à stocker, on procède en 2 niveaux.

- Au premier niveau, en utilisant une fonction aléatoire h de $\mathcal{H}$, on répartit les clés en m sous-ensembles S_j, formés de n_j clés ayant même valeur de hachage j, pour $j = 0..m-1$.
  (Le nombre total de clés est $n = \sum_{j=0}^{m-1} n_j$).
- Au second niveau, on crée pour chaque S_j, une taille de hachage de taille n_j^2, via une fonction de hachage $h^{(j)}$, choisie aléatoirement dans $\mathcal{H}$.

1. Décrire l'algorithme précédent en pseudo-code.
2. Montrer que la taille de la mémoire requise par cet algorithme est en $O(n)$.
   (On pourra montrer – ou admettre – que la valeur moyenne de $\sum_{j=0}^{m-1} n_j^2$ est inférieure à $2n$).

Exercice 4.2 ***: Hachage*** k***-universel***

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

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

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

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

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

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

Question 4.2.1 Soit U un univers ayant n clés et $\mathcal{H}$ la famille de toutes les fonctions de U dans $[0, 1, \ldots, m-1]$. Montrer que $\mathcal{H}$ est k-universelle (pour $k \leq n$).

Question 4.2.2 Écrire la définition de famille 2-universelle.

Question 4.2.3 Montrer que si une famille $\mathcal{H}$ de fonctions de hachage est 2-universelle alors elle vérifie, pour toute clé x et pour toute valeur v dans $[0, 1, \ldots, m-1]$: $\Pr(h(x) = v) = \frac{1}{m}$.

Question 4.2.4 Montrer que si une famille $\mathcal{H}$ de fonctions de hachage est 2-universelle alors elle est universelle.

Question 4.2.5 Soit $U = \{x_1, x_2, x_3, x_4\}$, on considère les familles $\mathcal{H}_0$ et $\mathcal{H}_1$ de fonctions de U dans $\{0, 1\}$ données 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 $\mathcal{H}_1$?

Question 4.2.6 Une famille universelle est-elle nécessairement 2-universelle ?

Question 4.2.7 Montrer que si une famille $\mathcal{H}$ de fonctions de hachage est $(k+1)$-universelle alors elle est k-universelle.

Question 4.2.8 En déduire que si une famille $\mathcal{H}$ de fonctions de hachage est k-universelle, avec $k \geq 2$, alors elle est universelle et elle vérifie, pour toute clé x et pour toute valeur v dans $[0, 1, \ldots, m-1]$: $\Pr(h(x) = v) = \frac{1}{m}$.

5 Filtres de Bloom

Exercice 5.3 ***: Appartenance à un ensemble***

Étant donné un ensemble A de n éléments, l'objectif est de stocker les éléments de A avec peu de mémoire, pour ensuite tester l'appartenance à A de façon probabiliste.

La structure est un tableau T de m bits, et on utilise k fonctions de hachage uniformes et indépendantes $h_1, \ldots, h_k$ à images dans $\{0, \cdots, m-1\}$. Pour "placer" un élément x de A dans cette structure, on met la valeur 1 dans chaque $T[h_i(x)]$, pour $i = 1, \ldots, k$.

Fonction `Consruction` (ensemble A, entier m, fonctions de hachage $h_1, \ldots, h_k$)
T = table de m bits, initialisée à 0
PourChaque a_i dans A
  PourChaque fonction de hachage h_j
    $T[h_j(a_i)] = 1$
  FinPour
FinPour
Retourne T

Ensuite, pour rechercher si un élément y appartient à l'ensemble, on calcule tous les $h_i(y)$ et on vérifie que tous les bits correspondant dans T valent 1.

Fonction `Appartient?` (élément y, table T , fonctions de hachage $h_1, \ldots, h_k$)
PourChaque fonction de hachage h_j
  Si $T[h_j(y)] = 0$ **Alors Retourne** NON
FinPour
Retourne OUI

Lorsque la fonction `Appartient?` retourne NON, c'est que l'élément y n'est pas dans A mais il est possible qu'elle retourne OUI pour un élément qui n'appartient pas à l'ensemble (on parle alors de *faux positif*).

Question 5.3.1 Construire la table T pour $A = \{9, 11\}$, avec $m = 5$, $k = 2$, et les fonctions de hachage $h_1(x) = x \mod 5$ et $h_2(x) = 2 \times x + 3 \mod 5$.
Appliquer ensuite la fonction `Appartient?` pour $y = 15$ et $y = 16$. Expliquer le résultat.

Question 5.3.2 Étant données les tables T_A et T_B de taille m, associées à 2 ensembles A et B, en utilisant les mêmes fonctions de hachage $h_1, ..., h_k$. Expliquer comment construire la table associée à l'union $A \cup B$.

Question 5.3.3 Dans la suite on considère une table T_A associée à un ensemble A.
Montrer que la probabilité pour qu'un bit donné de la table soit égal à 0 est $p_0 = (1 - 1/m)^{kn}$.
En déduire la probabilité p_r que la fonction `Appartient?` donne un faux positif est majorée par $(1 - \exp(-kn/m))^k$, pour m grand (on utilisera l'approximation $(1 - 1/m)^{kn} \sim \exp(-kn/m)$).

Question 5.3.4 Étant donné un ensemble de n éléments, on peut jouer sur la taille de la table et le nombre de fonctions de hachage pour minimiser la probabilité de faux positifs.
Pour m et n fixés, montrer que la valeur de k qui minimise la probabilité p_r est $\frac{m}{n} \log 2$ (pour m grand).
Discuter du choix des valeurs de k et m.