TD 4 : Tables de hachage

Exercice 1. *Un hachage sans collision*

Une fonction de hachage $h : U \to \{0, \dots, m-1\}$ est *sans collision* pour un ensemble $X \subset U$ si pour tout $x, y \in X$, $h(x) \neq h(y)$. Dans cet exercice, on suppose X fixé.

1. Donner une condition nécessaire et suffisante sur X pour qu'il existe une fonction de hachage sans collision pour X.
2. Supposons qu'on ait choisi une fonction h aléatoire. Exprimer l'espérance du nombre de collisions pour X en fonction de m et $n = |X|$.
3. Quelle est la probabilité qu'une fonction aléatoire h soit sans collision pour X.
4. Suposons qu'on cherche une fonction sans collision pour X en tirant des fonctions aléatoires tant qu'on en a pas trouvé une qui convienne. Quelle est l'espérance du nombre de tirages nécessaires ?

Exercice 2. *Case la plus remplie*

Soit $h : U \to \{0, \dots, n-1\}$ une fonction de hachage aléatoire. On insère n clefs dans une table T de taille n à l'aide de h, en utilisant une résolution par chaînage. On souhaite connaître l'espérance de la case de T la plus remplie.

1. **i.** Soit j un indice entre 0 et $n-1$. Quelle est l'espérance du nombre d'élément en case j ?
   ii. Pourquoi on ne peut pas conclure directement ?
2. Soit X_j la variable aléatoire qui compte le nombre d'éléments en case $T_{[j]}$.
   i. Montrer que $\Pr\left[X_j \geq k\right] \leq \binom{n}{k}\frac{1}{n^k}$.
   ii. Montrer que $\binom{n}{k} \leq \frac{n^k}{k!}$.
   iii. Montrer que $k! \geq \sqrt{k^k}$ pour tout $k \geq 0$. *Indications. En utilisant* $1 + x \leq e^x$ *(pour tout x), montrer que* $(1 + \frac{1}{k})^k \leq 1 + k$ *pour tout* $k > 0$*, puis que* $(k+1)^k \leq (k+1)k^k$ *pour* $k \geq 0$*. En déduire par récurrence que* $(k!)^2 \geq k^k$.
   iv. Déduire des questions précédentes que $\Pr\left[X_j \geq k\right] \leq \frac{1}{k^{k/2}}$.
3. On pose $k = \frac{c \log n}{\log \log n}$, pour une certaine constante c.
   i. Justifier que $\frac{c \log n}{\log \log n} \geq \sqrt{\log n}$ pour n suffisamment grand.
   ii. En déduire que pour n suffisamment grand, $\frac{1}{k^{k/2}} \leq \frac{1}{n^{c/4}}$.
   iii. En déduire que $\Pr\left[X_j \geq k\right] \leq \frac{1}{n^{c/4}}$.
4. **i.** Montrer que $\Pr\left[\max_j X_j \geq k\right] \leq n \Pr\left[X_j \geq k\right]$.
   ii. En déduire que la probabilité que la case la plus remplie possède plus de $c \log n / \log \log n$ éléments est $\leq 1/n^d$ pour une constante d à déterminer.
5. On note M le nombre d'élément dans la case la plus remplie, et on veut borner $\mathbb{E}[M]$.
   i. Montrer que pour tout k, $\mathbb{E}[M] \leq k \Pr[M \leq k] + n \Pr[M > k]$.
   ii. En déduire que $\mathbb{E}[M] = O(\log n / \log \log n)$.

Exercice 3. *Une famille quasi-universelle*

On s'intéresse à la famille de fonctions de hachage $\mathscr{H}_{w,\ell} = \{h_a : a \in I_w\}$ où I_w est l'ensemble des entiers impairs entre 0 et $2^w - 1$, et h_a est définie par

$$h_a(x) = \left\lfloor \frac{ax \bmod 2^w}{2^{w-\ell}} \right\rfloor$$

pour tout $x \geq 0$. La notation $ax \bmod 2^w$ représente le reste dans la division euclidienne de ax par 2^w.
Un entier positif est représenté par un tableau de bits, sa *taille* d'un entier est la taille de ce tableau.

1. On veut montrer que $h_a(x) \in \{0, \dots, 2^\ell - 1\}$ pour tout $x \geq 0$.
   i. Si a et x, positifs, sont de taille w, borner la valeur de l'entier ax et en déduire sa taille.

ii. Identifier $ax \bmod 2^w$ dans le tableau représentant ax.
iii. Identifier $h_a(x)$ dans le tableau représentant ax.
iv. Conclure.

2. Écrire une implantation Python efficace de h_a sous la forme d'une fonction `h(a,x,l,w)` qui utilise la multiplication, la soustraction et les opérations sur les bits (`&`, `<<`, `>>`) mais pas de division euclidienne (ni `%` ni `//`) ou de puissance.

3. **i.** Montrer que pour tout x, $a \in I_w$, $ax \bmod 2^w$ appartient également à I_w.
ii. Montrer que pour x, a, $b \in I_w$, $a \neq b$ implique $ax \bmod 2^w \neq bx \bmod 2^w$.
iii. En déduire que pour tout x, $y \in I_w$, il existe un unique $a \in I_w$ tel que $ax \bmod 2^w = y$.

4. On va montrer que la famille $\mathscr{H}_{w,\ell}$ est quasi-universelle. On fixe pour cela deux entiers positifs $x < y < 2^w$.

i. Soit $a \in I_w$. Montrer que $h_a(x) = h_a(y)$ si et seulement si $h_a(y-x) = 0$ ou $h_a(y-x) = 2^\ell - 1$.

On écrit $(y-x) \bmod 2^w$ sous la forme $q2^r$ où q est impair.

ii. Montrer que les $r+1$ bits de poids faible de $(aq2^r) \bmod 2^w$ sont $10\cdots0$.
iii. Montrer que $h_a(y-x)$ est constitué des ℓ bits de poids fort de $aq2^r \bmod 2^w$.
iv. Montrer que si $r+\ell > w$, $\Pr[h_a(y-x) = 0] = \Pr[h_a(y-x) = 2^\ell - 1] = 0$.
v. Montrer que si $r+\ell < w$, $\Pr[h_a(y-x) = 0] = \Pr[h_a(y-x) = 2^\ell - 1] = 1/2^\ell$.
vi. Montrer que si $r+\ell = w$, $\Pr[h_a(y-x) = 0] = 0$ et $\Pr[h_a(y-x) = 2^\ell - 1] = 2/2^\ell$.
vii. En déduire que pour tout $x \neq y$, $\Pr[h_a(x) = h_a(y)] \leq 2/2^\ell$.

Exercice 4. *Filtres de Bloom*

On s'intéresse dans cet exercice à une structure de données qui permet de stocker de manière très compressée un ensemble (statique, c'est-à-dire duquel on ne supprime jamais d'élément). La contrepartie est la présence de faux-positifs : la structure de données répond parfois que x appartient à l'ensemble alors que ça n'est pas le cas. Son utilisation en pratique vient en appui d'une *vraie* structure de donnée, pour fournir un pré-test d'appartenance très rapide [1].

Un filtre de Bloom pour un ensemble de taille n est donné par un entier m (la taille de la représentation) et k fonctions de hachage $h_1, \ldots, h_k$ indépendantes. Un ensemble X est représenté par un mot booléen w de taille m. L'ensemble vide est représenté par le mot $0\cdots0$. Pour insérer un nouvel élément x, on passe à 1 les k bits de w d'indices $h_1(x), \ldots, h_k(x)$. Un bit peut être mis plusieurs fois à 1. Pour tester si un élément y appartient à x, on vérifie si $w_{h_j(y)}$ vaut 1 pour $1 \leq j \leq k$: si c'est le cas, on répond « oui » et sinon on répond « non ».

Dans la suite, on suppose qu'on a construit la représentation w d'un ensemble X de taille n. On se place dans le modèle aléatoire pour les fonctions de hachage.

1. Laquelle des deux réponses de l'algorithme de recherche est toujours exacte ?
2. Montrer que le i-ème bit w_i de w vaut 1 si et seulement s'il existe $x \in X$ et j tels que $h_j(x) = i$.
3. Quelle est la probabilité p que le i-ème bit de w soit égal à 0 ? *On rappelle qu'on se place dans le modèle aléatoire, et que la probabilité dépend du choix des fonctions de hachage.*

On fait maintenant l'hypothèse qu'une fraction p des bits de w sont à 0.

4. Pourquoi cette hypothèse ne découle pas de la question précédente ?
5. Soit $y \notin X$. Quelle est la probabilité d'obtenir un faux-positif, c'est-à-dire que l'algorithme de recherche réponde « oui » sur l'entrée y ?
6. Montrer qu'en prenant $k = m \ln 2/n$, cette probabilité est exponentiellement petite. *On pourra utiliser, entre autres, que* $1 - x \geq e^{-2x}$ *pour* $x \leq 1/2$.

Exercice 5. *Adressage ouvert*

On suppose qu'on dispose d'une table de hachage T de taille m, contenant n éléments. Les conflits sont résolus par *adressage ouvert* : on dispose de m fonctions de hachages $h_0, \ldots, h_{m-1}$ et un élément x est inséré en case $T[h_0(x)]$ si elle est libre, sinon en case $T[h_1(x)]$ si elle est libre, et ainsi de suite. *On suppose l'hypothèse forte de hachage uniforme : pour tout* x, $(h_0(x), h_1(x), \ldots, h_{m-1}(x))$ *est une permutation aléatoire de* $\{0, \ldots, m-1\}$, *et si* $x \neq y$, $h_i(x)$ *est indépendant de* $h_j(y)$ *pour tout* i *et tout* j.

On effectue une recherche *infructueuse* : on cherche un élément x dans la table mais il n'y est pas. On souhaite borner l'espérance $\mathbb{E}_{m,n}$ du nombre de cases visitées lors de cette recherche.

1. Voir `https://en.wikipedia.org/wiki/Bloom_filter#Examples` pour de nombreux exemples d'utilisation de ces objets en pratique.

1. Montrer que pour tout nouvel élément x, la probabilité que $T[h_0(x)]$ soit libre est $1-n/m$.
2. Montrer que $\mathbb{E}_{m,n} = 1 + \frac{n}{m}\mathbb{E}_{m-1,n-1}$.
3. En déduire que $\mathbb{E}_{m,n} \leq m/(m-n)$.
4. On note X la variable aléatoire qui compte le nombre de cases visitées lors d'une recherche infructueuse. On vient de montrer que $\mathbb{E}[X] = \mathbb{E}_{m,n} \leq m/(m-n)$. On souhaite maintenant borner $\Pr[X \geq k]$ pour un k fixé. Pour cela, on définit pour tout j^l'évènement E_j : « les j premières cases visitées sont occupées ».
   i. Exprimer l'évènement « $X \geq k$ » en fonction de $E_1, \ldots, E_{k-1}$, pour $k \geq 2$.
   ii. En déduire que $\Pr[X \geq k] = \Pr[E_{k-1} | E_1 \wedge E_2 \wedge \cdots \wedge E_{k-2}] \Pr[X \geq k-1]$, pour $k \geq 2$.
   iii. Montrer que pour tout $j > 1$, $\Pr\left[E_j \middle| E_1 \wedge \cdots \wedge E_{j-1}\right] = \frac{n-j+1}{m-j+1}$.
   iv. En déduire que $\Pr[X \geq k] \leq (n/m)^{k-1}$ pour $1 \leq k \leq m$.

On imagine maintenant qu'on part de la table vide (de taille m) et qu'on insère successivement n valeurs, avec $n \leq m/2$. On rappelle qu'une insertion doit trouver la première case vide parmi les cases d'indices $h_0(x), \ldots, h_{m-1}(x)$: cette recherche est l'équivalent d'une recherche infructueuse.

5. On note X_i le nombre de cases visitées lors de la $i^{\text{ème}}$ insertion, et $X = \max_{1 \leq i \leq n} X_i$.
   i. Montrer que pour tout i, $\Pr[X_i > k] < 1/2^k$.
   ii. En déduire que pour tout i, $\Pr[X_i > 2\log n] < 1/n^2$.
   iii. Montrer que $\Pr[X > 2\log n] < 1/n$.
   iv. En déduire que l'espérance de X est $O(\log n)$.
   Écrire $\mathbb{E}[X] = \sum_{k \leq 2\log n} k \Pr[X = k] + \sum_{k > 2\log n} k \Pr[X = k]$ *et borner chacune des deux sommes.*