4I505 CPA – Examen de rattrapage Juin 2016

Les documents distribués dans le cadre du cours et les notes personnelles sont autorisés.

1 Famille de fonctions de hashage 2-universelle

Soit $\mathcal{H}_p$ une famille de fonctions de hashage de $\{1,\dots,p-1\}$ vers $\{1,\dots,p-1\}$, où $p \geq 3$ est un nombre premier. $\mathcal{H}_p$ est constituée de toutes les $p-1$ fonctions h_a de la forme

$$h_a(x) = x \cdot a \pmod{p},$$

où a est un entier de $1,\dots,p-1$ et $\cdot$ est la multiplication habituelle des entiers, ici modulo p. Par exemple pour $p=5$, $\mathcal{H}_k$ est composée des 4 fonctions dont les images sont les suivantes (l'entrée en ligne a et colonne x est $h_a(x)$)

$a\backslash x$	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

Est-ce que cette famille est 2-universelle ?

2 Entrées-sorties d'un musée

Un portique d'un musée relève toutes les entrées et sorties des visiteurs. Il génère un flux de couples (d,x) avec $d \in \{\mathrm{IN}, \mathrm{OUT}\}$ et x un identifiant dans $\{1,\dots,N\}$. Chaque visiteur a un identifiant unique x et quand il entre dans le musée le couple (IN, x) est généré, quand il quitte le musée le couple (OUT, x) est généré.

Vous recevez un flux de taille $2n-1$, correspondant à l'entrée et la sortie de n visiteurs distincts, mais il y a un visiteur d'identifiant x_0 qui est seulement entré dans le musée sans en être sorti. Le flux donné est tel que pour chaque $x \neq x_0$, si (IN, x) apparaît dans le flux, le couple correspondant (OUT, x) apparaît aussi dans le flux à une position ultérieure.

1. Proposez un algorithme qui lit ce flux et retourne x_0. Cet algorithme devrait lire le flux en une seule passe, et utiliser une mémoire de taille $O(\log N)$.

2. Le portique vient de se casser et désormais ne peut plus distinguer les entrées des sorties. Il génère alors un flux d'entiers correspondant aux identifiants des visiteurs sans distinguer s'il s'agit d'une entrée ou d'une sortie. Vous recevez un flux de taille $2n-1$ avec la propriété que chaque entier x y apparaît soit 0 soit 2 fois, sauf un entier x_0 qui y apparaît 1 fois. Proposez un algorithme qui lit ce flux et qui détermine x_0. Sa complexité en mémoire doit être $O(\log N)$.

3 Échantillonnage proportionnel

On vous donne un flot de $n > 0$ triplets de la forme (a, t, d) où a est une adresse IP sur 32 bits (par exemple 129.4.15.1) et $t, d > 0$ des entiers avec $t + d < T$ pour une constante T. Il s'agit de la trace journalière[1] d'un serveur web, un triplet (a, t, d) indiquant que la machine d'adresse a était connectée au serveur entre les temps t et $t + d$. Le temps est compté en millisecondes depuis minuit et T est le nombre de millisecondes en une journée.

Notons par $D[a]$ la somme des durées d sur tous les triplets (a, t, d) du flux. Concevez un algorithme de flux à une seule passe qui génère au hasard une adresse a avec une probabilité proportionnel à $D[a]$. Quelle est sa complexité en mémoire ?

Par exemple sur le flux $(10.0.0.1, 10, 2), (10.0.0.2, 19, 5), (10.0.0.1, 32, 2), (10.0.0.1, 41, 3)$ les probabilités de génération devraient être les suivants.

adresse	probabilité
10.0.0.1	0.4
10.0.0.2	0.5
10.0.0.3	0.1

[1] pour un jour fixée