ENS LYON
L3 Probabilités

Probabilités, L3 Informatique Fondamentale

**Devoir maison numéro 1 – Correction**
A rendre pour le 13 mars.

**Exercice 1 Bits aléatoires I**

Voici trois algorithmes naturels pour tirer au sort un élément $k$ de loi uniforme dans $\{1,2,3,4,5\}$ à partir d'une suite de bits aléatoires $(X_n)_{n\geqslant 1}$ (supposés i.i.d. de loi $\mathsf{B}(1/2)$).

A. On tire au sort un élément uniformément dans $\{1,\dots,8\}$ en utilisant 3 bits, et on recommence si l'élément n'est pas dans $\{1,\dots,5\}$.

B. On interprète $(X_n)_{n\geqslant 1}$ comme le développement en base 2 d'un élément $x$ de l'intervalle $[0,1]$ et on renvoie $(k+1)$ si $x \in [k/5,(k+1)/5[$.

C. On considère les 5 premiers bits. S'ils ne sont pas tous identiques, on choisit $k$ parmi les indices des éléments minoritaires (s'il y en a deux, cela nécessite un bit supplémentaire). S'ils ont tous identiques, on répète la procédure sur les 5 bits suivants. [1]

Pour chacun des algorithmes, calculer l'espérance du nombre de bits nécessaires pour générer un élément de loi uniforme dans $\{1,\dots,5\}$. Pouvez-vous faire plus économique ?

On note $E_A$, $E_B$ et $E_C$ les espérances.

A. On a $E_A = 3 + \frac{3}{8}E_A$ d'où on tire $E_A = 4,8$ bits.

B. Après 3 bits, avec probabilité 1/2 on a déterminé l'intervalle correspondant (si $\mathtt{000},\mathtt{010},\mathtt{101},\mathtt{111}$) et avec probabilité 1/2 il y a deux intervalles possibles ; dans le cas le nombre de tirages supplémentaires suit une loi géométrique de paramètre 1/2, qui a pour espérance 2. On a donc $E_B = 4$ bits.

C. Avec probabilité 1/16 on a utilisé 5 bits et on doit répéter ; avec probabilité 5/16 on termine en utilisant 5 bits et avec probabilité 5/8 on termine en utilisant 6 bits. On a donc $E_C = (E_C + 5)/16 + \frac{5}{16}5 + \frac{5}{8}6$ donc $E_C = 6$ bits.

Voici comment améliorer l'algorithme A. Dans le cas où le nombre obtenu est $> 5$, on utilise un bit supplémentaire ; parmi les 6 situations équiprobables que cela concerne, dans 5 cas on renvoie un des 5 nombres $\{1,2,3,4,5\}$ et dans le dernier cas on recommence tout au début. L'espérance vérifie

$$E = \frac{5}{8}3 + \frac{5}{16}4 + \frac{1}{16}(4+E)$$

et $E = 3,6$ bits.

1. Algorithme utilisé dans les cours de récréation sous le nom de «main noire, main blanche»

**Exercice 2 Bits aléatoires II**

On s'intéresse au problème suivant : étant donnée une suite $(X_n)$ de variables aléatoires i.i.d. de loi de Bernoulli $\mathsf{B}(p)$ pour $0 < p < 1$ (paramètre inconnu), produire une suite $(Y_k)$ de variables aléatoires i.i.d. de loi de Bernoulli $\mathsf{B}(1/2)$.

L'algorithme le plus simple consiste à observer les termes de $(X_n)$ deux par deux et à produire $(Y_k)$ selon la règle

$$\mathtt{00} \to \Lambda,\ \mathtt{01} \to \mathtt{0},\ \mathtt{10} \to \mathtt{1},\ \mathtt{11} \to \Lambda$$

où $\Lambda$ est le mot vide. Cette procédure génère des bits aléatoires avec une espérance de $\frac{2p(1-p)}{2} = p(1-p)$ bit produits par bit lu : pour deux bits biaisés lus, on produit 1 bit débiaisé avec probabilité $2p(1-p)$.

1. Considérons l'algorithme qui consiste à observer les termes de $(X_n)$ quatre à quatre et à produire $(Y_k)$ selon la règle

$$\mathtt{0000} \to \Lambda,\ \mathtt{1111} \to \Lambda$$

$$\mathtt{0001} \to \mathtt{00},\ \mathtt{0010} \to \mathtt{01},\ \mathtt{0100} \to \mathtt{10},\ \mathtt{1000} \to \mathtt{11}$$

$$\mathtt{1110} \to \mathtt{00},\ \mathtt{1101} \to \mathtt{01},\ \mathtt{1011} \to \mathtt{10},\ \mathtt{0111} \to \mathtt{11}$$

$$\mathtt{0011} \to \mathtt{00},\ \mathtt{0101} \to \mathtt{01},\ \mathtt{0110} \to \mathtt{10},\ \mathtt{1001} \to \mathtt{11},\ \mathtt{1010} \to \mathtt{0},\ \mathtt{1100} \to \mathtt{1}$$

En quel sens cet algorithme produit-il des bits aléatoires non biaisés ? Calculer l'espérance du nombre de bits produits par bit lu et montrer qu'elle est supérieure à $p(1-p)$.

Soit $m$ le mot renvoyé par l'algorithme. Pour $l = 1$ ou $l = 2$, conditionnellement à $|m| = l$, on peut vérifier que la loi de $m$ est uniforme sur l'ensemble des $2^l$ mots de longueur $l$.

Puis, pour 4 bits lus, on produit :
— 0 bit avec proba $p^4 + (1-p)^4$,
— 1 bit avec proba $2p^2(1-p)^2$,
— 2 bits avec proba $4p(1-p)^3 + 4p^3(1-p) + 4p^2(1-p)^2$.
Donc l'espérance vaut

$$\begin{aligned}
\mathbf{E} &= \frac{1}{4} \times \left[2p^2(1-p)^2 \ + \ 2 \times \left(4p(1-p)^3 + 4p^3(1-p) + 4p^2(1-p)^2\right)\right] \\
&= \frac{5}{2}p^2(1-p)^2 \ + \ 2p(1-p) \times \left[(1-p)^2 + p^2\right] \\
&= p(1-p) \times \left[\frac{5}{2}p(1-p) \ + \ 2p^2 + 2(1-p)^2\right] \\
&= p(1-p)\left[2 - \frac{3}{2}p(1-p)\right]
\end{aligned}$$

et on a $p(1-p) \leqslant \frac{1}{4}$ donc $\frac{3}{2}p(1-p) \leqslant \frac{3}{8}$ , d'où finalement $\mathbf{E} > p(1-p)$.

2. Peut-on améliorer l'algorithme de la question précédente ?

L'algorithme où on regarde les $X_n$ quatre par quatre et celui où on les regarde deux par deux sont en fait des cas particuliers d'une famille d'algorithmes $(\mathcal{A}_m)$ qu'on va décrire ci-dessous.

Remarquons d'abord que pour tout $m$ et pour tout $1 \leqslant k < m$, il y a $\binom{m}{k}$ séquences de $m$ bits où 1 apparaît exactement $k$ fois, et elles sont toutes équiprobables de probabilité $p^k(1-p)^{m-k}$. On peut donc associer chacune de ces séquences à un entier $i \in \left\{0, \dots, \binom{m}{k} - 1\right\}$.

Pour tout $m \geqslant 2$, on définit $\mathcal{A}_m$ de la façon suivante :

On considère $(X_n)$ comme une suite de mots de $m$ bits. Pour tout mot, on fait :
— Si tous les bits sont identiques, on renvoie $\Lambda$.
— Sinon, il y exactement $k$ occurrences de 1 pour $1 \leqslant k < m$.
On associe la séquence à un entier $i \in \left\{0, \dots, \binom{m}{k} - 1\right\}$ avec la remarque précédente.
— Si $\log_2 i < \lfloor \log_2 \binom{m}{k} \rfloor$ on renvoie $i$ en binaire (codé sur $\lfloor \log_2 \binom{m}{k} \rfloor - 1$ bits).
— Sinon on renvoie $\Lambda$.

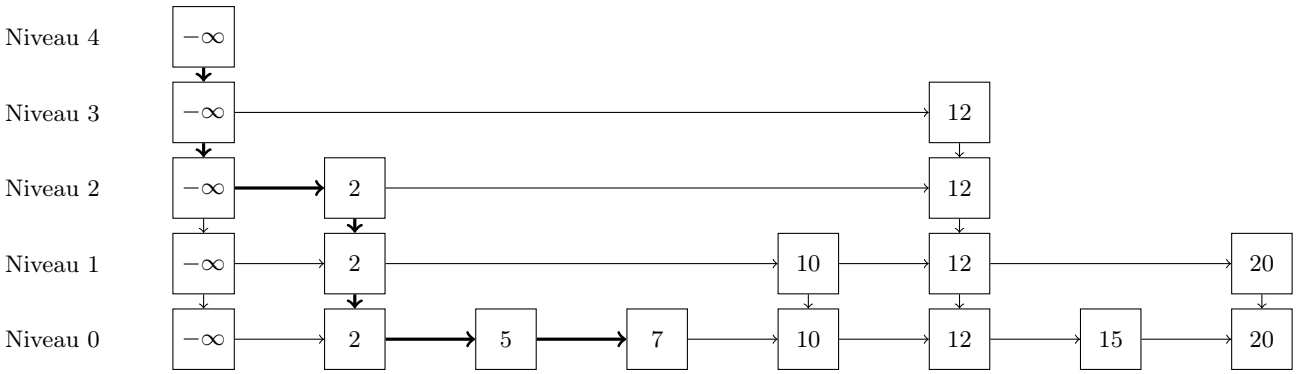
On peut étudier les $\mathcal{A}_m$ pour $m < 2$ pour améliorer l'espérance (les calculs sont similaires à ceux de la question précédente).

**Exercice 3 Liste à sauts aléatoire**

Soit $(X_n)$ une suite de variables aléatoires i.i.d. de loi géométrique de paramètre $1/2$.

1. On pose $M_n = \max(X_1, \dots, X_n)$. Montrer que, pour une constante $C$ à déterminer, on a
$$\lim_{n\to\infty} \mathbf{P}(M_n \geqslant C \log n) = 0.$$
On a $\mathbf{P}(X_i > k) = 2^{-k}$ et donc par la borne de l'union, $\mathbf{P}(M_n > k) = n2^{-k}$, qui tend vers 0 pour $k = C \ln n$ si $C > \frac{1}{\ln 2}$.
2. On pose $S_k = X_1 + \cdots + X_k$. Montrer en utilisant l'inégalité de Chernoff I que pour tout $\lambda > 0$
$$\mathbf{P}(S_k > (1+\lambda)2k) \leqslant \exp\left(-\frac{\lambda^2}{1+\lambda}k\right)$$
La variable aléatoire $S_k$ est la loi de l'indice du $k$ème 1 dans une suite de bits aléatoires $(Z_n)_{n\geqslant 1}$. On a donc $\mathbf{P}(S_k > N) = \mathbf{P}(Z_1 + \cdots + Z_N < k)$. Ainsi, si $N = (1+\lambda)2k$ :
$$\mathbf{P}(S_k > (1+\lambda)2k) = \mathbf{P}(Z_1 + \cdots + Z_N < k) = \mathbf{P}(Z_1 + \cdots + Z_N < \mu - a)$$
pour $\mu = (1+\lambda)k$ et $a = \lambda k$ ; l'inégalité découle de Chernoff I.

Une *liste à sauts* ou *skip list* est une structure de données stockant des éléments indexés par un ensemble de clés $L = \{x_1, \dots, x_n\}$ totalement ordonné : $x_1 < x_2 < \cdots < x_n$. On définit par récurrence une suite décroissante de sous-ensembles de $L$ de la façon suivante. On pose $L_0 = L$, puis on définit $L_{j+1} \subset L_j$ comme un sous-ensemble aléatoire de $L_j$ obtenu en conservant chaque élément avec probabilité $1/2$ (tous les choix étant indépendants) jusqu'à ce que $L_j = \emptyset$. On dit que $L_j$ est le niveau $j$ de la liste. Voici un exemple pour $L = \{2, 5, 7, 10, 12, 15, 20\}$.



3. Montrer que le nombre de niveaux est $O(\log n)$ avec grande probabilité.
On remarque que les hauteurs des différents éléments sont des v.a. i.i.d. de loi géométrique de paramètre 1/2. On peut donc appliquer la question 1.

4. On ajoute à chaque niveau un élément noté $-\infty$, inférieur à tous les autres. L'accès à un élément $x$ se fait en partant de l'élément $-\infty$ du dernier niveau et en itérant les opérations suivantes jusqu'à arriver à $x$ : si l'élément suivant dans le niveau actuel est $\leqslant x$, se déplacer vers ce dernier. Sinon, descendre d'un niveau. Dans l'exemple ci-dessus, les flèches en gras correspondent à l'accès à l'élément de clé 7. Montrer qu'avec grande probabilité, l'accès à tout élément se fait en $O(\log n)$ opérations.

   Soit $N$ le nombre de niveaux. Fixons un élément $x$. On doit majorer la longueur $\ell_x$ du chemin partant de l'élément $-\infty$ du dernier niveau jusqu'à l'élément $x$ du dernier niveau possible. $\ell_x$ est la somme du nombre de déplacements verticaux et du nombre de déplacements horizontaux. Pour $k \in \mathbb{N}$ et $t \in \mathbb{R}$, pour avoir $\ell_q > t$, soit il y a plus de $k$ niveaux ($N > k$), soit il y a moins de $k$ niveaux et le nombre de déplacements est $> t$. Maintenant, pour un élément d'un niveau donné, la longueur pour aller à un élément du chemin du niveau supérieur suit une loi géométrique de paramètre $\frac{1}{2}$ car pour chaque élément d'un niveau, il y a probabilité 1/2 de monter d'un niveau et probabilité 1/2 de rester au même niveau.

   Ceci permet de comparer $\ell_x$ à une somme de $N$ lois géométriques iid $X_i \sim \mathcal{G}(1/2)$.

   Il suit :

$$\begin{aligned}\mathbf{P}(\ell_x > t) &\leqslant \mathbf{P}(N > k) + \mathbf{P}((X_1 + \dots X_N > t) \cap (N \leqslant k)) \\ &\leqslant \mathbf{P}(N > k) + \mathbf{P}((X_1 + \dots X_k > t))\end{aligned}$$

Pour $t = (1+\lambda)2k$, $\lambda > 0$ quelconque, on a par la question 2 :

$$\begin{aligned}\mathbf{P}(\ell_x > (1+\lambda)2k) &\leqslant \mathbf{P}(N > k) + \mathbf{P}((X_1 + \dots X_k > (1+\lambda)2k)) \\ &\leqslant \mathbf{P}(N > k) + \exp\left(-\frac{\lambda^2}{1+\lambda}k\right).\end{aligned}$$

Il vient par la question 1 :

$$\mathbf{P}(\ell_x > (1+\lambda)2k) \leqslant n2^{-k} + \exp\left(-\frac{\lambda^2}{1+\lambda}k\right).$$

On veut $\lim_n \mathbf{P}(\max_x \ell_x > c\log n) = 0$, or par la borne de l'union :

$$\mathbf{P}\left(\max_x \ell_x > c\log n\right) \leqslant \sum_x \mathbf{P}\left(\ell_x > c\log n\right).$$

On veut donc choisir $k$ et $\lambda$ tel que

$$n \times \left(n2^{-k} + \exp\left(-\frac{\lambda^2}{1+\lambda}k\right)\right) \to_{n\to\infty} 0.$$

On prend $k = C\ln n$ avec $C > 2/\ln 2$ et $\lambda > 1$, et donc par la borne de l'union

$$\mathbf{P}\left(\max_x \ell_x > (1+\lambda)2k\right) \leqslant 2/n,$$

ce qui montre que l'accès à tout élément se fait en temps $(1+\lambda)2k = O(\log n)$ avec grande probabilité.