ENS LYON
L3 Probabilités

Probabilités, L3 Informatique Fondamentale

Devoir maison numéro 1

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 $\{1, \ldots, 8\}$ en utilisant 3 bits, et on recommence si l'élément n'est pas dans $\{1, \ldots, 5\}$.

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 $\{1, \ldots, 5\}$. Pouvez-vous faire plus économique ?

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 Λ 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 $p(1 - p)$.

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

---

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

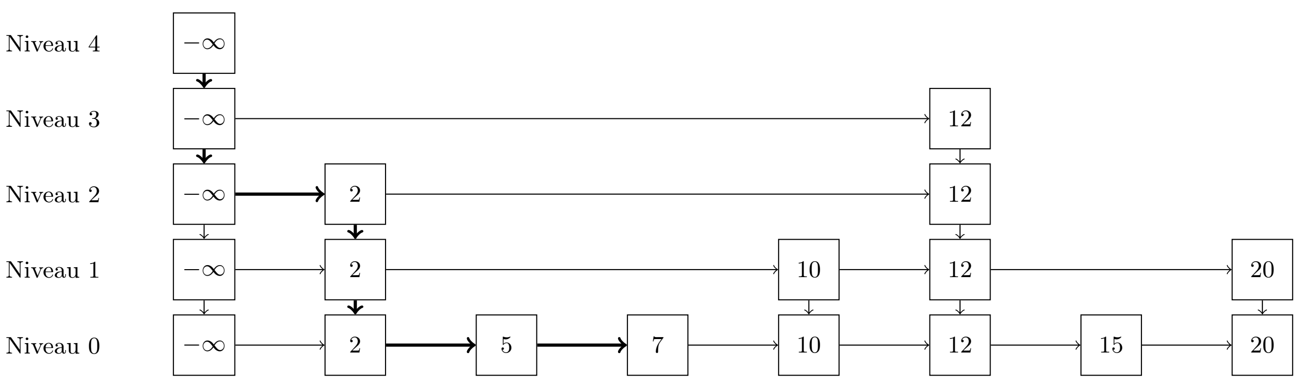
Exercice 3 Liste à sauts aléatoire

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
$$\lim_{n\to\infty} \mathbf{P}(M_n \geqslant C \log n) = 0.$$

2. On pose $S_k = X_1 + \cdots + X_k$. Montrer en utilisant l'inégalité de Chernoff I que pour tout $\lambda > 0$
$$\mathbf{P}(S_n > (1+\lambda)2k) \leqslant \exp\left(-\frac{\lambda^2}{1+\lambda}k\right)$$

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.

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.