

DEVOIR MAISON 1

Exercice 1

Collectionneur de coupons

Un individu collectionne les images dans les boîtes de céréales. Chaque boîte en contient une et une seule, et il y a n images différentes au total.

L'image contenue dans une boîte est choisie uniformément parmi les n images, et indépendamment des autres boîtes.

On note X_i la variable aléatoire du nombre de boîtes ouvertes jusqu'à l'obtention de i images différentes, et X la variable aléatoire du nombre de boîtes à ouvrir au total (donc $X = X_n$).

1. Que vaut X_1 ? Quelle est la distribution de $X_i - X_{i-1}$ pour $2 \leq i \leq n$?
2. Combien de boîtes faut-il acheter en moyenne pour obtenir toutes les images? (on donnera aussi l'ordre de grandeur asymptotique quand n est grand).

On note $H(n) = \sum_{i=1}^n \frac{1}{i}$.

3. Quelle est la probabilité que le temps pour collectionner les n coupons soit le double au moins de cette espérance? Borner $\mathbf{P}(X \geq 2nH(n))$ à l'aide des inégalités de Markov et de Tchebychev.

Exercice 2

Distribution uniforme

Considérons un flot de données $n_1, n_2, \dots, n_t$. Une fois reçu ce flot, on veut générer de manière uniforme un nombre parmi les w derniers reçus ($P(X = k)$ est proportionnelle au nombre de k reçus parmi les w derniers nombres) On suppose que w n'est pas connu à l'avance.

1. Comment faire avec une mémoire de $O(t)$ nombres (on ne tient pas compte de la place pour coder un entier)?

Le but de cet exercice est de montrer que l'on peut faire beaucoup mieux en moyenne. Soient $X_1, X_2, \dots, X_n$ n variables aléatoires uniformément distribuées sur $[0, 1]$. On admet que deux v.a. uniformément distribuées sur $[0, 1]$ ont même valeur avec probabilité 0.

2. Quelle est la probabilité que $X_1 = \min(X_1, \dots, X_t)$?
3. Montrer que l'algorithme suivant permet de générer un nombre de manière uniforme parmi les w derniers reçus.

Algorithme 1 : query(t, w)

début

 pour $s = 1$ à t faire

$v(s) \leftarrow n_s$;

$r(s) \leftarrow \text{random}[0, 1]$;

$j \leftarrow \text{Argmin}_{i \in [t-w+1, t]} r(i)$;

 retourner $v(j)$.

fin

4. Améliorer cet algorithme en ne gardant que l'information nécessaire. Quelle est alors le nombre moyen de valeurs à mémoriser?