Examen du 10 janvier 2012 (Durée 2 heure)

Une feuille manuscrite recto-verso est autorisée, les autres documents ne sont pas autorisés.
Il sera tenu compte de la rigueur de la rédaction et de la clarté de la présentation

Pour tout l'examen on suppose que l' on dispose d'un générateur aléatoire `Random()` qui fournit une séquence de nombres réels aléatoires indépendants de même loi uniforme sur $[0, 1[$.

Algorithme inconnu...

Génère(n)
  p =Random() (*)
  $S = 0$
  pour $i = 1$ **à** n **faire**
    si Random()$<$ p **alors**
      $S = S + 1$
    fin si
  fin pour
  Retourne S

Question 1.1 : Loi de S sachant p

Supposant la valeur de p fixée ligne (*), quelle est la loi de la valeur retournée par le programme **Génère**(n) ?

Question 1.2 : Loi de S

En utilisant les fonctions génératrices calculer la loi de la valeur retournée par le programme. En déduire un algorithme plus simple de simulation.

Générateur poly-exponentiel

On veut générer une variable aléatoire X de densité sur $\mathbb{R}^+$: $f(x) = xe^{-x}$.
Question 2.1 : Algorithme de génération

Écrire un algorithme de génération de X et calculer son coût moyen. *On pourra utiliser la méthode de rejet avec la densité xe^{-x}, mais ce n'est pas la seule méthode.*

Question 2.2 : Somme de variables aléatoires de loi exponentielle

Soit Y_1 et Y_2 variables aléatoires indépendantes de même loi exponentielle de paramètre 1. Calculer la loi de $Y = Y_1 + Y_2$ et en déduire un nouvel algorithme de génération de X.

Question 2.3 : Généralisation (bonus)

Proposer un algorithme de dénération d'une variable aléatoire de densité

$f(x) = P(x)e^{-x}$ avec $P(x)$ polynôme à coefficients positifs.

Comparison de codes...

On dispose de 2 programmes A et B réalisant le même calcul. On teste les deux programmes sur un même ensemble de jeux de données (par exemple générées uniformément) et on mesure les temps d'exécution respectifs de ces programmes.

Sur 900 expériences le programme A a été plus rapide 520 fois.
Question 3.1 :

Peut-on en déduire que le programme A est significativement meilleur que B ?

Question 3.2 :

A-t-on intérêt à utiliser le programme A ? Si non détailler les expériences ou les tests nécessaires afin de mieux décider.

Chapeaux et soirée arrosée... *(6 points)*

À une soirée chacun des n invités laisse son chapeau au vestiaire, le préposé au vestiaire, un peu farceur, rend les chapeaux dans un ordre farfelu.

On s'intéresse à la probabilité que chacun des invités reparte avec un autre chapeau que le sien. On note d_n le nombre de permutations de n objets sans point fixe.

Question 4.1 : n petit

Donner toutes les possibilités pour $n = 1, 2, 3, 4$, et déduire les valeurs de d_1, d_2, d_3, d_4.

$d_1 = 0$, (un seul invité qui repart avec son chapeau)
$d_2 = 1$, (une seule façon de rendre les chapeaux sans que personne ne reparte avec son chapeau)
$d_3 = 2$, $(2, 3, 1)$ et $(3, 1, 2)$,
$d_4 = 9$ $2143, 3412, 4321, 4123, 3412, 2341, 4132, 3142, 2431$

Question 4.2 : Équation de récurrence d_n (2 pas)

Établir l'équation de récurrence ci-dessous à l'aide d'un argument combinatoire.

$$d_{n+1} = n(d_n + d_{n-1}) \tag{1}$$

Question 4.3 : Équation de récurrence (1 pas)

Déduire que

$$d_n = nd_{n-1} + (-1)^n,$$

et la formule

$$d_n = n!\left(\frac{1}{0!} - \frac{1}{1!} + \frac{1}{2!} - \cdots + \frac{(-1)^n}{n!}\right).$$

Question 4.4 : Ordre des chapeaux uniforme

On suppose que les chapeaux sont rendu de manière aléatoire, uniformément sur l'ensemble des permutations possibles des chapeaux

1. Quelle est la probabilité qu'un invité (le premier par exemple) reparte avec son chapeau ?
2. Calculer la probabilité p_n qu'aucun invité ne reparte avec son chapeau.
3. Calculer la limite p de p_n pour n grand et calculer une borne de l'erreur $|p_n - p|$.
4. Donner p_{13} avec une précision de 10^{-2} [1].

Question 4.5 :

Calculer le nombre moyen d'invités repartant avec leur chapeau. *Indication : Considérer la variable aléatoire X_i indiquant si l'invité i repart avec son chapeau ou non*)

Question 4.6 : Générateur de mélange

1. Écrire un algorithme de génération uniforme de permutations de n chapeaux.
2. Écrire un algorithme basé sur la méthode de rejet qui génère uniformément une permutation des chapeaux telle qu'acun invité ne reparte avec son chapeau. Calculer le coût moyen de cette génération.
3. A partir de l'équation (1), écrire un générateur uniforme de permutation sans point fixe.

1. $e \simeq 2.71828$