Mathématiques

Devoir surveillé n° 6

Lycée THIERS
Année 23-24

1BCPST 2

30 mars 2024
Durée : 3h

Qualité de la rédaction, clarté des raisonnements, présentation, orthographe et ponctuation font partie des critères de notation.
Il est vivement recommandé de lire attentivement les questions et d'*encadrer les résultats*. L'usage des calculatrices est ***interdit***.

Exercice 1. *Série de jeux sur deux machines*

Un joueur décide de jouer aux machines à sous. Il va jouer sur deux machines $\mathcal{A}$ et $\mathcal{B}$ qui sont réglées de la manière suivante :

- la probabilité de gagner sur la machine $\mathcal{A}$ est $3/10$;
- la probabilité de gagner sur la machine $\mathcal{B}$ est $2/10$.

Comme le joueur ne sait pas laquelle des deux machines est la plus favorable, il décide de procéder de la manière suivante :

- il commence par choisir une machine au hasard ;
- après chaque partie, il change de machine si il a perdu et il rejoue sur la même machine si il a gagné.

Pour $k \in \mathbb{N}^*$ on définit les événements :

G_k : "le joueur gagne la k-ième partie",
A_k : "la k-ième partie se déroule sur la machine $\mathcal{A}$".

1. **Probabilités élémentaires**

   (a) Déterminer la probabilité de gagner la première partie.

   (b) Déterminer la probabilité que les trois premières parties se fassent sur la machine $\mathcal{A}$.

   (c) Calculer la probabilité de jouer la deuxième partie sur la machine $\mathcal{A}$.

   (d) Déterminer la probabilité de gagner la deuxième partie.

   (e) Sachant que la deuxième partie a été gagnée, quelle est la probabilité quelle ait été réalisée sur la machine $\mathcal{A}$?

   (f) Les événements G_1 et G_2 sont-ils indépendants ?

2. **Probabilité de gagner la k-ième partie**

   (a) Exprimer $\mathbb{P}(G_k)$ en fonction de $\mathbb{P}(A_k)$ uniquement.

   (b) On note $a_k = \mathbb{P}(A_k)$. Montrer que $a_{k+1} = -\frac{1}{2}a_k + \frac{4}{5}$.

   (c) En déduire une expression de a_n puis de $\mathbb{P}(G_n)$ en fonction de $n \in \mathbb{N}^*$.

   (d) Calculer la limite de $\mathbb{P}(G_n)$ puis interpréter cette limite.

   (e) Pour $n \in \mathbb{N}^*$ on pose $\tau_n = \dfrac{1}{n}\displaystyle\sum_{k=1}^{n} \mathbb{P}(G_k)$.

   On admettra que τ_n est la proportion moyenne de parties gagnées parmi les n premières.

   Calculer τ_n puis la limite de τ_n quand $n \to +\infty$. Donner une interprétation de cette limite.

3. **Probabilité de gagner tout le temps**
   (a) Calculer la probabilité de gagner les n premières parties avec la machine $\mathcal{B}$.
   (b) À partir de quelle valeur de n la probabilité de gagner les n premières parties avec la machine $\mathcal{B}$ est inférieure à $0,001$? On donne $\dfrac{\ln(0,002)}{\ln(5)} \approx -3,86$.
4. **Simulation informatique**
   (a) Simuler l'expérience aléatoire à l'aide d'une fonction python `simul` de paramètre n qui renvoie la liste des résultats des n premières parties sous la forme d'une n-liste de 0 et de 1. Le chiffre 1 signifie que la partie a été gagnée alors que 0 signifie qu'elle a été perdue.

   On propose de compléter le code suivant dans lequel `machine` désigne une variable qui peut prendre la valeur 1 si on utilise la machine A ou la valeur 0 si on utilise la machine B. La variable `issue` est un entier aléatoire qui va permettre de simuler une partie avec la machine A ou la machine B.

```
from random import randint
def simul(n):
    lst = ----- # on initialise la liste lst qui va contenir les résultats
    machine = randint(------,------) # on choisit une machine
    for k in range(------):
        issue = randint(-----,------)
        if machine == -----:
            if issue <= -----: # on simule une partie gagnée
                ----------------
            else:
                ------------------
                machine = ----- # on change de machine
        else:
            if issue <= --------:
                ---------------------
            else:
                --------------------
                machine = -----
    return --------
```

   (b) Écrire une fonction `approx1(n,p)` qui approche la probabilités de gagner exactement n parties lors des $2n$ premières parties avec p simulations.
   (c) Écrire une fonction `approx2(n,p)` qui approche la probabilités de gagner la $n^{\text{ème}}$ partie sachant que les précédentes ont été gagnées avec p simulations.
   (d) On rappelle que τ_n désigne $\dfrac{1}{n}\displaystyle\sum_{k=1}^{n} \mathrm{P}\left(G_k\right)$. Écrire une fonction `estim(n,p)` qui renvoie une approximation de τ_n avec p simulations.

   Indication. On pourra travailler avec une liste de compteurs `cpt` telle que `cpt[k-1]` soit le nombre de réalisations de l'événement G_k au cours des p simulations.

Exercice 2. *Algorithme de Babylone (1800 à 1600 avant J.C.)*

Méthode de Héron d'Alexandrie pour l'extraction de racines carrées (1er siècle ap. J.C.)

Les résultats établis dans cet exercice fournissent une méthode efficace utilisée par les ordinateurs et les machines à calculer pour approcher une racine carrée.

Soit $\alpha \in \mathbb{R}_+^*$ un paramètre réel strictement positif et soit la suite $(x_n)_{n\in\mathbb{N}}$ vérifiant :

$$(x_n)_n \begin{cases} x_0 = \alpha \\ \forall n \in \mathbb{N},\ x_{n+1} = \dfrac{x_n^2+\alpha}{2x_n} \end{cases}$$

Nous allons établir que la suite $(x_n)_n$ converge vers la racine carrée de α, $\sqrt{\alpha}$.

1. Montrer $\forall n \in \mathbb{N},\ x_n > 0$.
2. Développer $(x_{n-1} - \sqrt{\alpha})^2$ puis prouver que pour tout $n \in \mathbb{N}^*$: $x_n - \sqrt{\alpha} \geqslant 0$.
   On pourra exprimer x_n en fonction de x_{n-1}.
3. En déduire le signe de $x_{n+1} - x_n$ pour $n \in \mathbb{N}^*$, puis que la suite $(x_n)_n$ est décroissante à partir du rang 1 (c'est à dire que la suite extraite $(x_n)_{n\geqslant 1}$ est décroissante).
4. Prouver que la suite $(x_n)_n$ est convergente.
5. Après avoir justifié que $\lim\limits_{n\to+\infty} x_n$ est un réel positif, montrer que $\lim\limits_{n\to+\infty} x_n = \sqrt{\alpha}$.
6. (a) Écrire en `python` une fonction `suite_x()` : prenant deux paramètres : un nombre `a` et un entier `n`, l'appel de `suite_x(a,n)` retourne la valeur de x_n pour le paramètre $x_0 = a$.
   (b) Écrire en `python` une fonction `racine_carree()` prenant en paramètre un nombre `a`, qui retourne une valeur approchée de sa racine carrée. On pourra utiliser la fonction `suite_x()` du a).
   L'appel de `racine_carree(a)` :
   – retournera le calcul `suite_x(a, 10)` de x_{10} pour $x_0 = a$ lorsque $a > 0$,
   – retournera 0 lorsque $a = 0$
   – retournera un message d'erreur si $a < 0$.
   (c) Écrire une fonction python `affiche(a,n)` qui représente graphiquement les n premiers termes de la suite (x_n) de premier terme a et de la suite constante égale à $\sqrt{a}$.
   (d) Écrire une fonction python `seuil(a,p)` qui renvoie le plus petit n tel que $|x_n - \sqrt{a}| \leqslant p$.

Exercice 3. *Calcul de puissance de matrice*

On considère les matrices $A = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ et $U = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$.

1. Calculer U^2 puis conjecturer l'expression de U^k en fonction de U. Démontrer cette conjecture pour tout $k \in \mathbb{N}^*$. Cette relation est-elle vraie pour $k = 0$?
2. Exprimer A en fonction de I_3 et U et en déduire les coefficients de A^n pour tout $n \in \mathbb{N}$.

Exercice 4. *Dans une urne...*

On considère une urne contenant 5 boules bleues, 2 boules blanches et 5 boules rouges.
Les boules bleues sont numérotées de 1 à 5, les boules blanches sont numérotées de 1 à 2, les boules rouges sont numérotées de 1 à 5.
On tire simultanément 5 boules dans cette urne.

1. Combien y a-t-il de tirages en tout ?
2. Combien y a-t-il de tirages contenant les trois boules portant le numéro 1 ?
3. Combien y a-t-il de tirages contenant au moins une boule rouge ?
4. Combien y a-t-il de tirages contenant exactement 3 boules rouges ?
5. Combien y a-t-il de tirages d'une seule couleur ? bicolores ? tricolores ?
6. Combien y a-t-il de tirages sans boule blanche qui font apparaitre les cinq numéros 1, 2, 3, 4 et 5 ?