



PRÉPASIMPLE

prepasimple.fr

HEC — 2000

Sujet de mathématiques approfondies

Mathématiques approfondies — Prépa ECG

Annales de concours · Sujet

Le problème consiste en l'analyse d'un algorithme de tri et l'étude de sa complexité.

On note $\ln x$ le logarithme népérien d'un réel strictement positif x et $\frac{\ln(x)}{\ln(2)}$ son logarithme en base 2. On rappelle que :

$$\frac{\ln(x)}{\ln(2)} = \frac{\ln x}{\ln 2}$$

Partie I : Étude d'une suite réelle

Dans cette partie la lettre n désignera toujours un entier naturel au moins égal à 2.

On considère la suite $(u_k)_{k \in \mathbb{N}^*}$ définie par son premier terme u_1 et vérifiant, pour tout n , la relation de récurrence :

$$u_n = n - 1 + \frac{2}{n} \sum_{i=1}^{n-1} u_i$$

- 1) a) Calculer u_2 et u_3 en fonction de u_1 .
- b) Montrer que, pour tout n au moins égal à 3, on a : $nu_n - (n+1)u_{n-1} = 2n - 2$.
- 2) Pour tout entier naturel k non nul, on pose : $v_k = \frac{u_k}{k+1}$.
- a) Pour tout n au moins égal à 3, exprimer $v_n - v_{n-1}$ en fonction de n .
- b) Déterminer deux réels α et β vérifiant, pour tout réel x non nul et distinct de -1 , l'égalité :

$$\frac{2x-2}{x(x+1)} = \frac{\alpha}{x} + \frac{\beta}{x+1}$$

c) Pour tout n , établir l'égalité :

$$v_n = 2 \sum_{k=2}^n \frac{1}{k} + \frac{u_1}{3} - 2 + \frac{4}{n+1}$$

3) Pour tout n , on pose :

$$h_n = \sum_{k=2}^n \frac{1}{k}$$

et $z_n = \frac{1}{n} - \ln\left(\frac{n}{n-1}\right)$

a) Calculer u_n en fonction de h_n , u_1 et n .

b) Prouver l'égalité :

$$h_n = \sum_{k=2}^n z_k + \ln n.$$

c) Déterminer la nature de la série de terme général z_n .

d) En déduire un équivalent de h_n quand n tend vers l'infini.

e) Déterminer un équivalent de u_n quand n tend vers l'infini.

Partie II : Étude d'une suite de variables aléatoires

A. On considère un espace probabilisé dont la probabilité est notée P , une variable aléatoire Z , définie sur cet espace, prenant un nombre fini de valeurs réelles notées $z_1, z_2, \dots, z_p$ et un événement A de probabilité non nulle.

On note $E(Z/A)$ l'espérance de la variable aléatoire Z pour la probabilité conditionnelle sachant A , c'est-à-dire :

$$E(Z/A) = \sum_{i=1}^p z_i P([Z = z_i]/A)$$

Soit $(A_1, A_2, \dots, A_q)$ un système complet d'événements tous de probabilité non nulle. Prouver l'égalité :

$$E(Z) = \sum_{j=1}^q P(A_j) E(Z/A_j)$$

B. Toutes les variables aléatoires considérées dans cette sous-partie sont définies sur un même espace probabilisé dont la probabilité est notée P .

On considère une suite $(I_n)_{n \in \mathbb{N}^*}$ de variables aléatoires telle que, pour tout n non nul, I_n suit la loi uniforme sur l'ensemble $\{1, 2, \dots, n\}$ des entiers compris, au sens large, entre 1 et n .

D'autre part, on considère une suite $(X_n)_{n \in \mathbb{N}^*}$ de variables aléatoires ayant les propriétés suivantes :

- X_1 est la variable constante égale à 0;
- pour tout entier naturel n au moins égal à 2, les lois conditionnelles de X_n sachant $[I_n = 1]$ et de X_n sachant $[I_n = n]$ sont toutes deux égales à la loi de $n - 1 + X_{n-1}$;
- pour tout entier naturel n au moins égal à 3 et tout entier i tel que $2 \leq i \leq n - 1$, la loi conditionnelle de X_n sachant $[I_n = i]$ est égale à la loi de $n - 1 + Z_{n,i} + T_{n,i}$ où $Z_{n,i}$ et $T_{n,i}$ sont deux variables aléatoires indépendantes, $Z_{n,i}$ ayant même loi que X_{i-1} et $T_{n,i}$ ayant même loi que X_{n-i} .

Par exemple, on a :

$$P([X_6 = 9]/[I_6 = 1]) = P([X_5 = 4])$$

et aussi $P([X_6 = 9]/[I_6 = 3]) = P([Z_{6,3} + T_{6,3} = 4])$ ce qui, compte tenu des hypothèses, s'écrit :

$$P([X_6 = 9]/[I_6 = 3]) = \sum_j P([X_2 = j]) P([X_3 = 4 - j])$$

la somme étant étendue aux valeurs convenables de l'entier j .

1) a) Montrer que X_2 est une variable aléatoire presque sûrement constante égale à 1.

b) Établir les égalités : $P([X_3 = 2]) = \frac{1}{3}$ et $P([X_3 = 3]) = \frac{2}{3}$. Calculer l'espérance de X_3 qu'on notera U_3 .

2) Déterminer la loi de X_4 et calculer son espérance qu'on notera U_4 .

3) En procédant par récurrence, montrer que, pour tout entier naturel n non nul, X_n prend, presque sûrement, des valeurs entières inférieures ou égales à $\frac{n(n-1)}{2}$.

4) Soit n un entier naturel au moins égal à 2. On note U_n l'espérance de X_n .

a) À l'aide des résultats de la sous-partie A, établir l'égalité :

$$U_n = n - 1 + \frac{2}{n} \sum_{i=1}^{n-1} U_i$$

b) À l'aide de la partie I, donner l'expression de U_n en fonction de n ainsi qu'un équivalent de U_n quand n tend vers l'infini.

- 5) Pour tout entier naturel n non nul, on note α_n la plus petite valeur (entière) prise par la variable X_n avec une probabilité non nulle.

- a) Soit n et k deux entiers naturels, l'entier n étant au moins égal à 3.

Montrer que $P([X_n = k])$ est nul si et seulement si les nombres $P([n - 1 + X_{n-1} = k])$ et $P([n - 1 + Z_{n,i} + T_{n,i} = k])$ (l'entier i variant de 2 à $n - 1$) sont nuls.

En déduire que α_n est au moins égal au minimum des nombres $n - 1 + \alpha_{n-1}$, $n - 1 + \alpha_1 + \alpha_{n-2}$, $n - 1 + \alpha_2 + \alpha_{n-3}$, ..., $n - 1 + \alpha_{n-2} + \alpha_1$.

- b) On considère la fonction g définie, pour tout x strictement positif, par

$$g(x) = x \frac{\ln(x)}{\ln(2)} - 2x + 2.$$

- i) Montrer que g est convexe. Pour tout couple d'entiers (i, n) tel que $2 \leq i \leq n - 1$, en déduire l'inégalité :

$$g(i) + g(n + 1 - i) \geq 2g\left(\frac{n + 1}{2}\right)$$

- ii) Pour tout entier naturel n non nul, établir l'inégalité : $g(n + 1) - g(n) \leq \frac{\ln(n + 1)}{\ln(2)}$.
En traitant à part les cas $n = 1$ et $n = 2$, montrer que, pour tout entier naturel n non nul, on a : $g(n + 1) - g(n) \leq n - 1$.

- 6) En procédant par récurrence, établir, pour tout entier naturel n non nul, l'inégalité :

$$\alpha_n \geq (n + 1) \frac{\ln(n + 1)}{\ln(2)} - 2n$$

Partie III : Étude d'un algorithme de tri

- A. On considère un entier naturel n non nul et un ensemble $E = \{e_1, e_2, \dots, e_n\}$ où $e_1, e_2, \dots, e_n$ sont des réels vérifiant $e_1 < e_2 < \dots < e_n$. On munit l'ensemble des permutations de E de la probabilité uniforme notée P . On considère les n variables aléatoires $T_1, T_2, \dots, T_n$ qui, à toute permutation σ de E , associent les images des éléments de E par σ , c'est-à-dire $T_1(\sigma) = \sigma(e_1), T_2(\sigma) = \sigma(e_2), \dots, T_n(\sigma) = \sigma(e_n)$, et on note T le vecteur aléatoire $(T_1, T_2, \dots, T_n)$. Pour toute liste $(\alpha_1, \alpha_2, \dots, \alpha_n)$ d'éléments distincts de E on a donc :

$$P([T = (\alpha_1, \alpha_2, \dots, \alpha_n)]) = \frac{1}{n!}$$

- 1) Déterminer la loi de la variable aléatoire T_1 .
- 2) On suppose n au moins égal à 2. Pour toute permutation σ de E on note $T'_1(\sigma)$ le premier élément de la liste $(\sigma(e_1), \sigma(e_2), \dots, \sigma(e_n))$ inférieur à $\sigma(e_1) = T_1(\sigma)$ si un tel élément existe et $T'_1(\sigma) = 0$ sinon, $T'_2(\sigma)$ le deuxième élément inférieur à $\sigma(e_1)$ si un tel deuxième élément existe et $T'_2(\sigma) = 0$ sinon, etc., $T'_{n-1}(\sigma)$ le $(n-1)$ -ième élément inférieur à $\sigma(e_1)$ si un tel $(n-1)$ -ième élément existe et $T'_{n-1}(\sigma) = 0$ sinon.

Par exemple, si $n = 4$, $(e_1, e_2, e_3, e_4) = (3, 5, 7, 10)$ et

$(\sigma(e_1), \sigma(e_2), \sigma(e_3), \sigma(e_4)) = (7, 10, 5, 3)$ alors $T'_1(\sigma) = 5$, $T'_2(\sigma) = 3$ et $T'_3(\sigma) = 0$.

Soit k un entier vérifiant $1 \leq k \leq n-1$.

- a) Combien y a-t-il de listes $(i_1, i_2, \dots, i_k)$ d'entiers vérifiant $2 \leq i_1 < i_2 < \dots < i_k \leq n$?
- b) Soit $(\alpha_1, \alpha_2, \dots, \alpha_k)$ une liste d'éléments distincts de $\{e_1, e_2, \dots, e_k\}$. Établir l'égalité :

$$P\left(\bigcap_{j=1}^k [T'_j = \alpha_j] \cap [T_1 = e_{k+1}]\right) = \frac{1}{n k!}$$

c) En déduire l'égalité :

$$P \left(\bigcap_{j=1}^k [T'_j = \alpha_j] / [T_1 = e_{k+1}] \right) = \frac{1}{k!}$$

où $P(A/B)$ désigne la probabilité conditionnelle de A sachant B .

Ainsi la loi conditionnelle de $(T'_1, T'_2, \dots, T'_k)$ sachant $[T_1 = e_{k+1}]$ est uniforme.

B. Dans un programme écrit en langage Python, on représente un tableau de n nombres (où n est un entier naturel non nul fixé par l'utilisateur) par une liste T de longueur n , dont les éléments sont $T[0], T[1], \dots, T[n-1]$. Lorsqu'une liste est passée en argument à une fonction et que cette fonction modifie certains éléments de la liste, ces modifications subsistent après l'exécution de la fonction.

Soit T une liste qu'on suppose constituée de nombres distincts. Si deb et fin sont deux entiers tels que $0 \leq deb < fin \leq n - 1$, on dira qu'un élément $T[k]$ de la liste ($T[deb], T[deb+1], \dots, T[fin]$) est à sa place dans le sous-tableau $[T[deb], \dots, T[fin]]$ si les éléments $T[deb], T[deb+1], \dots, T[k-1]$ sont inférieurs à $T[k]$ et si les éléments $T[k+1], T[k+2], \dots, T[fin]$ sont supérieurs à $T[k]$ (bien sûr, si $k = deb$ ou $k = fin$, une seule de ces conditions subsiste).

Ainsi, si $n = 4$ et $T = [3, 1, 7, 5]$, alors $T[2]$ ($= 7$) est à sa place dans le sous-tableau $[T[0], T[1], T[2]]$ ($= [3, 1, 7]$) mais n'est pas à sa place dans le sous-tableau $[T[1], T[2], T[3]]$ ($= [1, 7, 5]$).

On dira que la liste T est triée si chacun de ses éléments est à sa place dans T .

On suppose qu'on dispose d'une fonction Python `placer(T, deb, fin)`, qui, si $fin \leq deb$, ne modifie pas T et renvoie deb , et qui, si $0 \leq deb < fin \leq n - 1$, effectue, à l'aide de $fin - deb$ comparaisons, les opérations suivantes :

i) d'une part, elle ne modifie que le sous-tableau $[T[deb], \dots, T[fin]]$ de sorte que les éléments de ce sous-tableau plus petits que $T[deb]$ « ont glissé » (sans permutation entre eux) à gauche de $T[deb]$ et les éléments de ce sous-tableau plus grands que

$T[\text{deb}]$ «ont glissé» (sans permutation entre eux) à droite de $T[\text{deb}]$. Ainsi $T[\text{deb}]$ se retrouve à sa place dans le sous-tableau modifié;

- ii) d'autre part, elle renvoie l'indice i tel que $T[i]$ reçoit, au cours de l'exécution de la fonction, la valeur qui était stockée dans $T[\text{deb}]$ avant cette exécution.

Par exemple, si $T = [12, 3, 8, 10, 6, 4, 5]$, l'instruction $pl = \text{placer}(T, 2, 5)$ une fois exécutée aura changé T en $[12, 3, 6, 4, 8, 10, 5]$ et affecté la valeur 4 à la variable pl , alors que l'instruction $pl = \text{placer}(T, 2, 3)$ aura laissé T inchangée et affecté la valeur 2 à la variable pl .

Par ailleurs on considère la fonction suivante, qui s'appelle elle-même (fonction récursive) et ne renvoie aucune valeur :

Python

```
def tri(T, deb, fin):  
    if fin > deb:  
        pl = placer(T, deb, fin)  
        if pl > deb:  
            tri(T, deb, pl - 1)  
        if pl < fin:  
            tri(T, pl + 1, fin)
```

- 1) a) L'entier i étant compris entre 0 et $n - 1$, quel est l'effet sur la liste T de l'instruction $\text{tri}(T, i, i)$?

- b) Dans cette sous-question on suppose qu'initialement $T = [2, 9, 6, 1, 5]$.

Déterminer la «trace» de l'instruction $\text{tri}(T, 0, 4)$ en donnant la liste des appels successifs des fonctions tri et placer (avec les valeurs de leurs paramètres) qui sont effectués et en indiquant à chaque fois les affectations de la variable pl et de la liste T .

- c) Expliquer succinctement l'effet et le principe de fonctionnement de la fonction tri en indiquant, en particulier, pourquoi l'algorithme s'arrête.

- 2) On se place à nouveau dans le contexte probabiliste de la sous-partie A et, si σ est une permutation de E , on affecte la valeur $[\sigma(e_1), \sigma(e_2), \dots, \sigma(e_n)]$ à la liste T , c'est-à-dire $T[0] = \sigma(e_1), T[1] = \sigma(e_2), \dots, T[n-1] = \sigma(e_n)$. On note $X_n(\sigma)$ le nombre de comparaisons faites lors des différentes exécutions de la fonction `placer` (et seulement au cours de celles-ci) quand on exécute l'instruction `tri(T, 0, n-1)`.
- a) À l'aide de la sous-partie A, montrer que la suite de variables aléatoires $(X_n)_{n \in \mathbb{N}^*}$ vérifie les hypothèses de II.B. En déduire un équivalent de l'espérance de X_n lorsque l'entier naturel n tend vers l'infini.
- b) Dans le cas où $E = \{1, 2, \dots, n\}$, donner (en le commentant de manière succincte) un exemple de liste nécessitant $\frac{n(n-1)}{2}$ comparaisons pour être triée.
- c) Dans le cas où $E = \{1, 2, \dots, 7\}$, donner de même un exemple de liste nécessitant 10 comparaisons pour être triée.
- d) Déterminer une suite d'entiers n pour lesquels, dans le cas où $E = \{1, 2, \dots, n\}$, il existe une liste d'éléments nécessitant $g(n+1)$ comparaisons pour être triée.