



PRÉPASIMPLE

prepasimple.fr

HEC — 2000

Corrigé de mathématiques approfondies

Mathématiques approfondies — Prépa ECG

Annales de concours · Corrigé

Corrigé

Partie I : Étude d'une suite réelle

Dans toute cette partie, n désigne un entier au moins égal à 2 et la relation

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

est valable pour tout $n \geq 2$.

1) a) Il s'agit d'appliquer la relation de récurrence avec $n = 2$ puis $n = 3$.

Pour $n = 2$:

$$u_2 = 2 - 1 + \frac{2}{2}u_1 = 1 + u_1$$

Pour $n = 3$, en utilisant la valeur de u_2 :

$$u_3 = 3 - 1 + \frac{2}{3}(u_1 + u_2) = 2 + \frac{2}{3}(2u_1 + 1) = \frac{8}{3} + \frac{4}{3}u_1$$

Conclusion

$$u_2 = 1 + u_1 \text{ et } u_3 = \frac{8}{3} + \frac{4}{3}u_1.$$

b) On écrit la relation de récurrence aux rangs n et $n - 1$, après multiplication par le dénominateur, puis on soustrait pour faire disparaître la somme.

Soit $n \geq 3$. Comme $n \geq 2$ et $n - 1 \geq 2$, la relation s'applique aux rangs n et $n - 1$:

$$nu_n = n(n - 1) + 2 \sum_{i=1}^{n-1} u_i$$

et

$$(n - 1)u_{n-1} = (n - 1)(n - 2) + 2 \sum_{i=1}^{n-2} u_i$$

Par différence, les termes $u_1, \dots, u_{n-2}$ disparaissent :

$$nu_n - (n - 1)u_{n-1} = (n - 1)(n - (n - 2)) + 2u_{n-1} = 2(n - 1) + 2u_{n-1}$$

d'où $nu_n - (n - 1)u_{n-1} - 2u_{n-1} = 2n - 2$.

Conclusion

pour tout $n \geq 3$, $nu_n - (n + 1)u_{n-1} = 2n - 2$.

2) a) On divise la relation de la question 1) b) par $n(n + 1)$ afin de faire apparaître

$$v_n = \frac{u_n}{n + 1} \text{ et } v_{n-1} = \frac{u_{n-1}}{n}.$$

Soit $n \geq 3$. Comme $n(n + 1) \neq 0$:

$$\frac{u_n}{n + 1} - \frac{u_{n-1}}{n} = \frac{2n - 2}{n(n + 1)}$$

Conclusion

pour tout $n \geq 3$, $v_n - v_{n-1} = \frac{2n-2}{n(n+1)}$.

b) On cherche α et β tels que les deux fractions rationnelles coïncident sur $\mathbb{R} \setminus \{0, -1\}$.

Méthode 1 — Identification après réduction au même dénominateur.

Pour tout réel $x \notin \{0, -1\}$:

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

Il suffit donc que $\alpha + \beta = 2$ et $\alpha = -2$, soit $\alpha = -2$ et $\beta = 4$. Vérification : pour $x \notin \{0, -1\}$,

$$-\frac{2}{x} + \frac{4}{x+1} = \frac{-2(x+1) + 4x}{x(x+1)} = \frac{2x-2}{x(x+1)}$$

Méthode 2 — Multiplication par un dénominateur et passage à la limite.

Supposons l'égalité vraie pour tout $x \notin \{0, -1\}$. En la multipliant par x :

$$\frac{2x-2}{x+1} = \alpha + \frac{\beta x}{x+1}; \text{ en faisant tendre } x \text{ vers } 0 \text{ (les deux membres sont continus en } 0),$$

on obtient $\alpha = -2$. En la multipliant par $x+1$: $\frac{2x-2}{x} = \frac{\alpha(x+1)}{x} + \beta$; en faisant

tendre x vers -1 , on obtient $\beta = \frac{-4}{-1} = 4$. Ces valeurs nécessaires conviennent d'après la vérification de la méthode 1.

La méthode 1 est systématique; la méthode 2 donne directement chaque coefficient associé à un pôle simple et montre en outre l'unicité du couple (α, β) .

Conclusion

$$\alpha = -2 \text{ et } \beta = 4, \text{ c'est-à-dire, pour tout réel } x \notin \{0, -1\}, \quad \frac{2x-2}{x(x+1)} = -\frac{2}{x} + \frac{4}{x+1}.$$

c) D'après 2) a) et 2) b) appliquée à $x = k$, pour tout entier $k \geq 3$:

$$v_k - v_{k-1} = -\frac{2}{k} + \frac{4}{k+1}$$

Par ailleurs $v_2 = \frac{u_2}{3} = \frac{1+u_1}{3}.$

Méthode 1 — Sommation télescopique.

Soit $n \geq 3$. En sommant pour k allant de 3 à n , la somme

$$\sum_{k=3}^n (v_k - v_{k-1})$$

est télescopique et vaut $v_n - v_2$, donc :

$$v_n - v_2 = -2 \sum_{k=3}^n \frac{1}{k} + 4 \sum_{k=3}^n \frac{1}{k+1} = -2 \sum_{k=3}^n \frac{1}{k} + 4 \sum_{k=4}^{n+1} \frac{1}{k}$$

Or

$$\sum_{k=4}^{n+1} \frac{1}{k} = \sum_{k=3}^n \frac{1}{k} - \frac{1}{3} + \frac{1}{n+1},$$

donc :

$$v_n - v_2 = 2 \sum_{k=3}^n \frac{1}{k} - \frac{4}{3} + \frac{4}{n+1}$$

Avec $v_2 = \frac{1}{3} + \frac{u_1}{3}$ et

$$2 \sum_{k=3}^n \frac{1}{k} = 2 \sum_{k=2}^n \frac{1}{k} - 1 :$$

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

Pour $n = 2$, le second membre vaut $1 + \frac{u_1}{3} - 2 + \frac{4}{3} = \frac{1+u_1}{3} = v_2$: l'égalité est encore vraie.

Méthode 2 — Récurrence.

Pour $n \geq 2$, notons $\mathcal{P}(n)$ l'égalité à établir. On vient de vérifier $\mathcal{P}(2)$. Soit $n \geq 3$ tel que $\mathcal{P}(n-1)$ soit vraie. Alors :

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

donc $\mathcal{P}(n)$ est vraie.

La méthode 1 permet de trouver la formule ; la méthode 2 ne fait que vérifier une formule déjà connue, ce qui est plus rapide lorsque l'énoncé la donne.

Conclusion

pour tout $n \geq 2$,

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

3) a) Comme $u_n = (n+1)v_n$ et

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

la question 2) c) donne :

$$u_n = (n+1) \left(2h_n + \frac{u_1}{3} - 2 \right) + 4 = 2(n+1)h_n + \frac{(n+1)u_1}{3} - 2n + 2$$

Conclusion

pour tout $n \geq 2$, $u_n = 2(n+1)h_n + \frac{(n+1)u_1}{3} - 2n + 2$.

b) Pour tout $k \geq 2$, $\frac{k}{k-1} > 0$ et $\ln \left(\frac{k}{k-1} \right) = \ln k - \ln(k-1)$, donc

$z_k = \frac{1}{k} - (\ln k - \ln(k-1))$. En sommant pour k de 2 à n , la seconde somme est télescopique :

$$\sum_{k=2}^n z_k = h_n - \sum_{k=2}^n (\ln k - \ln(k-1)) = h_n - (\ln n - \ln 1) = h_n - \ln n$$

Conclusion

pour tout $n \geq 2$,

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

c) Il s'agit de montrer que la série

$$\sum_{n \geq 2} z_n$$

converge.

Méthode 1 — Équivalent et série de Riemann.

On a $\ln\left(\frac{n}{n-1}\right) = -\ln\left(1 - \frac{1}{n}\right)$, donc $z_n = \frac{1}{n} + \ln\left(1 - \frac{1}{n}\right)$. Le développement limité usuel $\ln(1+x) = x - \frac{x^2}{2} + o(x^2)$ au voisinage de 0, appliqué à $x = -\frac{1}{n}$ qui tend vers 0, donne :

$$\ln\left(1 - \frac{1}{n}\right) = -\frac{1}{n} - \frac{1}{2n^2} + o\left(\frac{1}{n^2}\right)$$

donc $z_n = -\frac{1}{2n^2} + o\left(\frac{1}{n^2}\right)$, c'est-à-dire $-z_n \sim \frac{1}{2n^2}$.

La série

$$\sum \frac{1}{n^2}$$

est une série de Riemann convergente ($2 > 1$), donc

$$\sum \frac{1}{2n^2}$$

converge. Les suites $(-z_n)$ et $\left(\frac{1}{2n^2}\right)$ sont équivalentes et la seconde est strictement positive, donc $-z_n > 0$ à partir d'un certain rang et le théorème de comparaison par équivalents des séries à termes positifs montre que

$$\sum (-z_n)$$

converge. Par linéarité,

$$\sum z_n$$

converge.

Méthode 2 — Encadrement par une intégrale et série télescopique.

Soit $k \geq 2$. La fonction $t \mapsto \frac{1}{t}$ est décroissante sur $[k-1, k]$, donc pour tout $t \in [k-1, k]$, $\frac{1}{k} \leq \frac{1}{t} \leq \frac{1}{k-1}$. Par croissance de l'intégrale sur ce segment de longueur 1 :

$$\frac{1}{k} \leq \int_{k-1}^k \frac{dt}{t} = \ln\left(\frac{k}{k-1}\right) \leq \frac{1}{k-1}$$

Ainsi $0 \leq -z_k \leq \frac{1}{k-1} - \frac{1}{k}$. La série

$$\sum (-z_k)$$

est donc à termes positifs et ses sommes partielles vérifient :

$$\sum_{k=2}^n (-z_k) \leq \sum_{k=2}^n \left(\frac{1}{k-1} - \frac{1}{k} \right) = 1 - \frac{1}{n} \leq 1$$

Une série à termes positifs dont les sommes partielles sont majorées converge. Donc

$$\sum (-z_k)$$

converge, puis

$$\sum z_k$$

converge par linéarité, et sa somme appartient à $[-1, 0]$.

La méthode 1 fournit le comportement précis de z_n ; la méthode 2 évite tout développement limité et donne une majoration explicite de la somme, technique utile dès que le terme général compare une somme et une intégrale.

Conclusion

la série de terme général z_n est convergente.

d) Notons

$$S = \sum_{k=2}^{+\infty} z_k,$$

réel bien défini d'après 3) c). D'après 3) b), pour $n \geq 2$ (de sorte que $\ln n > 0$) :

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

La suite

$$\left(\sum_{k=2}^n z_k \right)_n$$

converge vers S , donc est bornée, et $\ln n \rightarrow +\infty$, donc

$$\frac{1}{\ln n} \sum_{k=2}^n z_k \rightarrow 0.$$

Ainsi $\frac{h_n}{\ln n} \rightarrow 1$.

Conclusion

$h_n \sim \ln n$ quand n tend vers l'infini (et plus précisément $h_n = \ln n + S + o(1)$).

e) On ne somme pas d'équivalents : on étudie directement le quotient de u_n par le candidat $2n \ln n$, non nul pour $n \geq 2$. D'après 3) a) :

$$\frac{u_n}{2n \ln n} = \frac{n+1}{n} \cdot \frac{h_n}{\ln n} + \frac{n+1}{n} \cdot \frac{u_1}{6 \ln n} + \frac{-2n+2}{2n \ln n}$$

Quand $n \rightarrow +\infty$: $\frac{n+1}{n} \rightarrow 1$, $\frac{h_n}{\ln n} \rightarrow 1$ d'après 3) d), $\frac{u_1}{6 \ln n} \rightarrow 0$ et

$$\frac{-2n+2}{2n \ln n} = \frac{-1 + \frac{1}{n}}{\ln n} \rightarrow 0.$$

Donc $\frac{u_n}{2n \ln n} \rightarrow 1$.

Conclusion

quelle que soit la valeur de u_1 , $u_n \sim 2n \ln n$ quand n tend vers l'infini.

Partie II : Étude d'une suite de variables aléatoires

A. On décompose chaque probabilité $P([Z = z_i])$ à l'aide de la formule des probabilités totales, puis on échange deux sommes finies.

La variable Z prend un nombre fini de valeurs, donc elle admet une espérance et

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

Comme $(A_1, \dots, A_q)$ est un système complet d'événements de probabilités non nulles, la formule des probabilités totales donne, pour tout $i \in \{1, \dots, p\}$:

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

Donc, en échangeant les deux sommes finies :

$$\begin{aligned} E(Z) &= \sum_{i=1}^p z_i \sum_{j=1}^q P(A_j) P([Z = z_i]/A_j) = \sum_{j=1}^q P(A_j) \sum_{i=1}^p z_i P([Z = z_i]/A_j) \\ &= \sum_{j=1}^q P(A_j) E(Z/A_j) \end{aligned}$$

Conclusion

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

Remarque utilisée en II.B.4) : le même calcul reste valable si Z prend presque sûrement ses valeurs dans un ensemble fini $\{z_1, \dots, z_p\}$. En effet, les valeurs prises avec une probabilité nulle ne contribuent pas à $E(Z)$, et pour tout j ,

$P([Z \notin \{z_1, \dots, z_p\}] \cap A_j) \leq P([Z \notin \{z_1, \dots, z_p\}]) = 0$, donc ces valeurs ne contribuent pas non plus à $E(Z/A_j)$.

B. Pour tout $n \geq 1$, les événements $[I_n = 1], \dots, [I_n = n]$ forment un système complet d'événements, chacun de probabilité $\frac{1}{n} > 0$. On l'utilisera systématiquement avec la formule des probabilités totales. On rappelle aussi qu'une variable ayant même loi qu'une variable presque sûrement égale à une constante c est elle-même presque sûrement égale à c .

1) a) On applique la formule des probabilités totales au système complet $([I_2 = 1], [I_2 = 2])$.

Pour $n = 2$, les lois conditionnelles de X_2 sachant $[I_2 = 1]$ et sachant $[I_2 = 2]$ sont égales à la loi de $1 + X_1$, c'est-à-dire de la constante 1. Donc

$P([X_2 = 1]/[I_2 = 1]) = P([X_2 = 1]/[I_2 = 2]) = 1$, et :

$$P([X_2 = 1]) = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 1 = 1$$

Conclusion

X_2 est presque sûrement constante égale à 1.

b) On applique la formule des probabilités totales au système complet $([I_3 = 1], [I_3 = 2], [I_3 = 3])$, chaque événement étant de probabilité $\frac{1}{3}$.

Sachant $[I_3 = 1]$ ou $[I_3 = 3]$, X_3 a la loi de $2 + X_2$, qui est presque sûrement égale à 3

d'après 1) a). Donc $P([X_3 = 3]/[I_3 = 1]) = P([X_3 = 3]/[I_3 = 3]) = 1$ et $P([X_3 = 2]/[I_3 = 1]) = P([X_3 = 2]/[I_3 = 3]) = 0$.

Sachant $[I_3 = 2]$ (ici $i = 2$ vérifie $2 \leq i \leq n - 1 = 2$), X_3 a la loi de $2 + Z_{3,2} + T_{3,2}$, où $Z_{3,2}$ et $T_{3,2}$ ont la loi de X_1 , donc sont presque sûrement nulles. Ainsi $2 + Z_{3,2} + T_{3,2} = 2$ presque sûrement, d'où $P([X_3 = 2]/[I_3 = 2]) = 1$ et $P([X_3 = 3]/[I_3 = 2]) = 0$.

Par la formule des probabilités totales :

$$P([X_3 = 2]) = \frac{1}{3}(0 + 1 + 0) = \frac{1}{3} \text{ et } P([X_3 = 3]) = \frac{1}{3}(1 + 0 + 1) = \frac{2}{3}$$

Ces deux probabilités ont pour somme 1, donc X_3 prend presque sûrement ses valeurs dans $\{2, 3\}$ et admet une espérance :

$$U_3 = 2 \cdot \frac{1}{3} + 3 \cdot \frac{2}{3} = \frac{8}{3}$$

Conclusion

$$P([X_3 = 2]) = \frac{1}{3}, \quad P([X_3 = 3]) = \frac{2}{3} \text{ et } U_3 = \frac{8}{3}.$$

2) On applique la formule des probabilités totales au système complet $([I_4 = i])_{1 \leq i \leq 4}$, chaque événement étant de probabilité $\frac{1}{4}$.

Sachant $[I_4 = 1]$ ou $[I_4 = 4]$: X_4 a la loi de $3 + X_3$, qui vaut 5 avec probabilité $\frac{1}{3}$ et 6 avec probabilité $\frac{2}{3}$ d'après 1) b).

Sachant $[I_4 = 2]$: X_4 a la loi de $3 + Z_{4,2} + T_{4,2}$ avec $Z_{4,2}$ de même loi que X_1 (nulle presque sûrement) et $T_{4,2}$ de même loi que X_2 (égale à 1 presque sûrement).

L'événement $[Z_{4,2} = 0] \cap [T_{4,2} = 1]$ est de probabilité 1 (son complémentaire est la réunion de deux événements négligeables), donc $3 + Z_{4,2} + T_{4,2} = 4$ presque sûrement.

Sachant $[I_4 = 3]$: de même, $Z_{4,3}$ a la loi de X_2 et $T_{4,3}$ celle de X_1 , donc $3 + Z_{4,3} + T_{4,3} = 4$ presque sûrement.

Par la formule des probabilités totales :

$$P([X_4 = 4]) = \frac{1}{4}(0 + 1 + 1 + 0) = \frac{1}{2}$$

$$P([X_4 = 5]) = \frac{1}{4} \left(\frac{1}{3} + 0 + 0 + \frac{1}{3} \right) = \frac{1}{6}$$

$$P([X_4 = 6]) = \frac{1}{4} \left(\frac{2}{3} + 0 + 0 + \frac{2}{3} \right) = \frac{1}{3}$$

Comme $\frac{1}{2} + \frac{1}{6} + \frac{1}{3} = 1$, X_4 prend presque sûrement ses valeurs dans $\{4, 5, 6\}$ et :

$$U_4 = 4 \cdot \frac{1}{2} + 5 \cdot \frac{1}{6} + 6 \cdot \frac{1}{3} = 2 + \frac{5}{6} + 2 = \frac{29}{6}$$

Conclusion

X_4 prend les valeurs 4, 5, 6 avec les probabilités respectives $\frac{1}{2}$, $\frac{1}{6}$, $\frac{1}{3}$, et $U_4 = \frac{29}{6}$.

- 3) Pour $m \geq 1$, notons $N_m = \frac{m(m-1)}{2}$ et $K_m = \{0, 1, \dots, N_m\}$. On démontre par récurrence forte la propriété, un peu plus forte que celle demandée, $\mathcal{H}(m)$:
 « $P([X_m \in K_m]) = 1$ », c'est-à-dire que X_m prend presque sûrement des valeurs entières comprises entre 0 et N_m . La minoration par 0 garantit que l'ensemble des valeurs possibles est fini.

Initialisation : $X_1 = 0$ et $K_1 = \{0\}$, donc $\mathcal{H}(1)$ est vraie ; $X_2 = 1$ presque sûrement et $K_2 = \{0, 1\}$, donc $\mathcal{H}(2)$ est vraie.

Hérédité : soit $n \geq 3$ tel que $\mathcal{H}(m)$ soit vraie pour tout $m \in \{1, \dots, n-1\}$. Par la formule des probabilités totales :

$$P([X_n \in K_n]) = \frac{1}{n} \sum_{i=1}^n P([X_n \in K_n] / [I_n = i])$$

Cas $i = 1$ ou $i = n$: $P([X_n \in K_n]/[I_n = i]) = P([n - 1 + X_{n-1} \in K_n])$. Par $\mathcal{H}(n - 1)$, presque sûrement $n - 1 + X_{n-1}$ est un entier compris entre $n - 1 \geq 0$ et $n - 1 + N_{n-1}$, et :

$$n - 1 + \frac{(n - 1)(n - 2)}{2} = \frac{(n - 1)(2 + n - 2)}{2} = \frac{n(n - 1)}{2} = N_n$$

Donc $P([n - 1 + X_{n-1} \in K_n]) = 1$.

Cas $2 \leq i \leq n - 1$: $P([X_n \in K_n]/[I_n = i]) = P([n - 1 + Z_{n,i} + T_{n,i} \in K_n])$. Comme $Z_{n,i}$ a la loi de X_{i-1} et $T_{n,i}$ celle de X_{n-i} , avec $1 \leq i - 1 \leq n - 2$ et $1 \leq n - i \leq n - 2$, les hypothèses $\mathcal{H}(i - 1)$ et $\mathcal{H}(n - i)$ montrent que l'événement

$B = [Z_{n,i} \in K_{i-1}] \cap [T_{n,i} \in K_{n-i}]$ est de probabilité 1 (son complémentaire est réunion de deux événements négligeables). Sur B , $n - 1 + Z_{n,i} + T_{n,i}$ est un entier au moins égal à $n - 1 \geq 0$ et au plus égal à $n - 1 + N_{i-1} + N_{n-i}$. En posant $a = i - 1$ et $b = n - i$, de sorte que $a + b = n - 1$:

$$\begin{aligned} N_{i-1} + N_{n-i} &= \frac{a(a - 1) + b(b - 1)}{2} = \frac{(a + b)^2 - (a + b) - 2ab}{2} = \frac{(n - 1)(n - 2)}{2} - ab \\ &\leq N_{n-1} \end{aligned}$$

Donc sur B , $n - 1 + Z_{n,i} + T_{n,i} \leq n - 1 + N_{n-1} = N_n$, et

$$P([n - 1 + Z_{n,i} + T_{n,i} \in K_n]) \geq P(B) = 1.$$

Finalement $P([X_n \in K_n]) = \frac{1}{n} \cdot n = 1$, ce qui établit $\mathcal{H}(n)$.

Conclusion

pour tout $n \geq 1$, X_n prend presque sûrement des valeurs entières comprises entre 0 et $\frac{n(n - 1)}{2}$, en particulier inférieures ou égales à $\frac{n(n - 1)}{2}$.

4) a) On applique le résultat de A à $Z = X_n$ et au système complet $([I_n = i])_{1 \leq i \leq n}$.

D'après 3), X_n prend presque sûrement ses valeurs dans l'ensemble fini K_n : elle admet une espérance et, d'après la remarque faite en A, la formule de A s'applique. On pose $U_1 = E(X_1) = 0$. Toutes les variables $X_m, Z_{n,i}, T_{n,i}$ prennent presque sûrement un

nombre fini de valeurs et admettent donc une espérance; deux variables de même loi ont même espérance.

L'espérance conditionnelle

$$E(X_n/[I_n = i]) = \sum_{k \in K_n} k P([X_n = k]/[I_n = i])$$

ne dépend que de la loi conditionnelle de X_n sachant $[I_n = i]$.

Si $i = 1$ ou $i = n$, cette loi est celle de $n - 1 + X_{n-1}$, donc par linéarité de l'espérance :

$$E(X_n/[I_n = i]) = E(n - 1 + X_{n-1}) = n - 1 + U_{n-1}$$

Si $2 \leq i \leq n - 1$ (ce qui suppose $n \geq 3$), cette loi est celle de $n - 1 + Z_{n,i} + T_{n,i}$, donc par linéarité :

$$E(X_n/[I_n = i]) = n - 1 + E(Z_{n,i}) + E(T_{n,i}) = n - 1 + U_{i-1} + U_{n-i}$$

La formule de A donne alors, avec $P([I_n = i]) = \frac{1}{n}$:

$$\begin{aligned} U_n &= \frac{1}{n} \left[2(n - 1 + U_{n-1}) + \sum_{i=2}^{n-1} (n - 1 + U_{i-1} + U_{n-i}) \right] = n - 1 \\ &+ \frac{1}{n} \left[2U_{n-1} + \sum_{i=2}^{n-1} U_{i-1} + \sum_{i=2}^{n-1} U_{n-i} \right] \end{aligned}$$

(si $n = 2$, les sommes sur i de 2 à $n - 1$ sont vides). Les changements d'indice $j = i - 1$ et $j = n - i$ donnent

$$\sum_{i=2}^{n-1} U_{i-1} = \sum_{i=2}^{n-1} U_{n-i} = \sum_{j=1}^{n-2} U_j,$$

donc :

$$U_n = n - 1 + \frac{1}{n} \left[2U_{n-1} + 2 \sum_{j=1}^{n-2} U_j \right] = n - 1 + \frac{2}{n} \sum_{i=1}^{n-1} U_i$$

Conclusion

pour tout $n \geq 2$,

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

(avec $U_1 = 0$).

b) La suite $(U_n)_{n \geq 1}$ vérifie la relation de récurrence de la partie I, avec premier terme $U_1 = 0$.

Soit (u_n) la suite de la partie I de premier terme $u_1 = 0$. Par récurrence forte, $U_n = u_n$ pour tout $n \geq 1$: c'est vrai pour $n = 1$, et si $U_i = u_i$ pour tout $i \leq n - 1$, alors

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

D'après I.3) a) avec $u_1 = 0$, pour tout $n \geq 2$:

$$U_n = 2(n+1) \sum_{k=2}^n \frac{1}{k} - 2n + 2$$

Vérification : on retrouve $U_2 = 6 \cdot \frac{1}{2} - 2 = 1$, $U_3 = 8 \cdot \frac{5}{6} - 4 = \frac{8}{3}$ et $U_4 = 10 \cdot \frac{13}{12} - 6 = \frac{29}{6}$, conformément à 1) et 2).

D'après I.3) e), $U_n \sim 2n \ln n$, soit encore $U_n \sim (2 \ln 2) n \frac{\ln(n)}{\ln(2)}$ avec $2 \ln 2 \approx 1,39$.

Conclusion

$$U_n = 2(n+1) \sum_{k=2}^n \frac{1}{k} - 2n + 2$$

et $U_n \sim 2n \ln n$ quand n tend vers l'infini.

5) D'après 3),

$$\sum_{k \in K_n} P([X_n = k]) = 1,$$

donc l'ensemble des entiers $k \in K_n$ tels que $P([X_n = k]) > 0$ est une partie finie non vide de $\mathbb{N}$; toute autre valeur est prise avec une probabilité nulle. Le nombre α_n est donc bien défini. D'après 1) et 2) : $\alpha_1 = 0, \alpha_2 = 1, \alpha_3 = 2, \alpha_4 = 4$.

a) Soit $n \geq 3$ et $k \in \mathbb{N}$. Par la formule des probabilités totales et les hypothèses sur les lois conditionnelles :

$$P([X_n = k]) = \frac{1}{n} \left[2 P([n-1 + X_{n-1} = k]) + \sum_{i=2}^{n-1} P([n-1 + Z_{n,i} + T_{n,i} = k]) \right]$$

Le crochet est une somme de nombres positifs affectés de coefficients strictement positifs : il est nul si et seulement si chacun de ces nombres est nul. Cela établit l'équivalence demandée.

Déduisons-en la minoration de α_n . Comme $P([X_n = \alpha_n]) > 0$, l'un au moins des nombres $P([n-1 + X_{n-1} = \alpha_n]), P([n-1 + Z_{n,i} + T_{n,i} = \alpha_n])$ (avec $2 \leq i \leq n-1$) est non nul.

Premier cas : $P([X_{n-1} = \alpha_n - n + 1]) > 0$. Alors l'entier $\alpha_n - n + 1$ est une valeur prise par X_{n-1} avec une probabilité non nulle, donc $\alpha_n - n + 1 \geq \alpha_{n-1}$ par définition de α_{n-1} , soit $\alpha_n \geq n - 1 + \alpha_{n-1}$.

Second cas : il existe $i \in \{2, \dots, n-1\}$ tel que $P([Z_{n,i} + T_{n,i} = m]) > 0$, où

$m = \alpha_n - n + 1$. Comme $Z_{n,i}$ prend presque sûrement ses valeurs dans l'ensemble fini K_{i-1} , et par indépendance de $Z_{n,i}$ et $T_{n,i}$:

$$\begin{aligned} P([Z_{n,i} + T_{n,i} = m]) &= \sum_{j \in K_{i-1}} P([Z_{n,i} = j] \cap [T_{n,i} = m - j]) \\ &= \sum_{j \in K_{i-1}} P([Z_{n,i} = j]) P([T_{n,i} = m - j]) \end{aligned}$$

Cette somme de termes positifs est non nulle, donc il existe un entier j tel que $P([X_{i-1} = j]) > 0$ et $P([X_{n-i} = m - j]) > 0$ (en utilisant l'égalité des lois). Par définition de α_{i-1} et α_{n-i} : $j \geq \alpha_{i-1}$ et $m - j \geq \alpha_{n-i}$, donc $m \geq \alpha_{i-1} + \alpha_{n-i}$, soit $\alpha_n \geq n - 1 + \alpha_{i-1} + \alpha_{n-i}$.

Lorsque i décrit $\{2, \dots, n-1\}$, les nombres $n - 1 + \alpha_{i-1} + \alpha_{n-i}$ sont exactement $n - 1 + \alpha_1 + \alpha_{n-2}, n - 1 + \alpha_2 + \alpha_{n-3}, \dots, n - 1 + \alpha_{n-2} + \alpha_1$.

Conclusion

dans tous les cas, α_n est supérieur ou égal à l'un des nombres $n - 1 + \alpha_{n-1}, n - 1 + \alpha_1 + \alpha_{n-2}, \dots, n - 1 + \alpha_{n-2} + \alpha_1$, donc à leur minimum.

b) i) La fonction g est de classe C^2 sur $]0, +\infty[$ comme produit et somme de fonctions de classe C^2 , et $g(x) = \frac{x \ln x}{\ln 2} - 2x + 2$. Pour tout $x > 0$:

$$g'(x) = \frac{\ln x + 1}{\ln 2} - 2 = \frac{\ln(x)}{\ln(2)} + \frac{1}{\ln 2} - 2 \text{ et } g''(x) = \frac{1}{x \ln 2} > 0$$

Comme $g'' \geq 0$ sur l'intervalle $]0, +\infty[$, la fonction g est convexe (et même g' est strictement croissante).

Soit (i, n) un couple d'entiers tel que $2 \leq i \leq n - 1$. Alors $i \geq 2$ et $n + 1 - i \geq 2$ sont strictement positifs, et leur moyenne vaut $\frac{n+1}{2}$.

Méthode 1 — Inégalité de convexité.

Par définition de la convexité, avec les coefficients $\frac{1}{2}$ et $\frac{1}{2}$:

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

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

Méthode 2 — Position par rapport à une tangente.

Une fonction convexe et dérivable sur un intervalle est au-dessus de ses tangentes. Avec $c = \frac{n + 1}{2} > 0$, pour tout $x > 0$, $g(x) \geq g(c) + g'(c)(x - c)$. En appliquant cette inégalité en $x = i$ et en $x = n + 1 - i$ et en additionnant :

$$g(i) + g(n + 1 - i) \geq 2g(c) + g'(c)(i + (n + 1 - i) - 2c) = 2g(c)$$

car $i + (n + 1 - i) = 2c$.

La méthode 1 utilise directement la définition ; la méthode 2 est commode dès que les points sont répartis symétriquement autour de c (les termes en $g'(c)$ se compensent) et se généralise à des sommes de plus de deux termes de moyenne c .

Conclusion

$$\text{pour } 2 \leq i \leq n - 1, \quad g(i) + g(n + 1 - i) \geq 2g\left(\frac{n + 1}{2}\right).$$

ii) Soit $n \geq 1$. On établit d'abord $g(n + 1) - g(n) \leq \frac{\ln(n + 1)}{\ln(2)}$. On utilisera l'inégalité

$\frac{1}{\ln 2} < 2$, qui équivaut à $\ln 2 > \frac{1}{2}$, c'est-à-dire à $2 > e^{1/2}$, soit $4 > e$ (les fonctions $\ln$ et $\exp$ étant strictement croissantes) : elle est vraie car $e < 3$.

Méthode 1 — Égalité des accroissements finis.

La fonction g est continue sur $[n, n+1]$ et dérivable sur $]n, n+1[$ (car $n \geq 1 > 0$). Il existe donc $c \in]n, n+1[$ tel que $g(n+1) - g(n) = g'(c)$. Comme g' est croissante, $g'(c) \leq g'(n+1)$, d'où :

$$g(n+1) - g(n) \leq \frac{\ln(n+1)}{\ln(2)} + \frac{1}{\ln 2} - 2 < \frac{\ln(n+1)}{\ln(2)}$$

Méthode 2 — Calcul direct et inégalité $\ln(1+x) \leq x$.

On a

$$\begin{aligned} g(n+1) - g(n) &= (n+1) \frac{\ln(n+1)}{\ln(2)} - n \frac{\ln(n)}{\ln(2)} - 2 = \frac{\ln(n+1)}{\ln(2)} \\ &+ n \left(\frac{\ln(n+1)}{\ln(2)} - \frac{\ln(n)}{\ln(2)} \right) - 2, \end{aligned}$$

et :

$$n \left(\frac{\ln(n+1)}{\ln(2)} - \frac{\ln(n)}{\ln(2)} \right) = \frac{n}{\ln 2} \ln \left(1 + \frac{1}{n} \right) \leq \frac{n}{\ln 2} \cdot \frac{1}{n} = \frac{1}{\ln 2} < 2$$

en utilisant l'inégalité de concavité $\ln(1+x) \leq x$ pour $x > -1$. Donc

$$g(n+1) - g(n) < \frac{\ln(n+1)}{\ln(2)}.$$

La méthode 1 n'utilise que la monotonie de g' et s'adapte à toute fonction convexe ; la méthode 2 est un calcul explicite qui précise l'écart.

Montrons maintenant que $g(n+1) - g(n) \leq n-1$ pour tout $n \geq 1$.

Cas $n = 1$: $g(1) = 0 - 2 + 2 = 0$ et $g(2) = 2 - 4 + 2 = 0$, donc $g(2) - g(1) = 0 = n - 1$.

Cas $n = 2$: $g(3) - g(2) = 3 \frac{\ln(3)}{\ln(2)} - 6 + 2 - 0 = 3 \frac{\ln(3)}{\ln(2)} - 4$. Or $3 \frac{\ln(3)}{\ln(2)} \leq 5$ équivaut à $\frac{\ln(27)}{\ln(2)} \leq \frac{\ln(32)}{\ln(2)}$, c'est-à-dire à $27 \leq 32$, ce qui est vrai. Donc $g(3) - g(2) \leq 1 = n - 1$.

Cas $n \geq 3$: d'après ce qui précède, il suffit de montrer $\frac{\ln(n+1)}{\ln(2)} \leq n-1$, c'est-à-dire $n+1 \leq 2^{n-1}$. Par récurrence sur $n \geq 3$: pour $n = 3$, $4 \leq 4$; si $n+1 \leq 2^{n-1}$, alors $n+2 \leq 2^{n-1} + 1 \leq 2^{n-1} + 2^{n-1} = 2^n$.

Les cas $n = 1$ et $n = 2$ doivent être traités à part car $\frac{\ln(n+1)}{\ln(2)} > n-1$ pour ces valeurs ($\frac{\ln(2)}{\ln(2)} = 1 > 0$ et $\frac{\ln(3)}{\ln(2)} > 1$).

Conclusion

pour tout $n \geq 1$, $g(n+1) - g(n) \leq \frac{\ln(n+1)}{\ln(2)}$ et $g(n+1) - g(n) \leq n-1$.

6) On remarque que

$$g(n+1) = (n+1) \frac{\ln(n+1)}{\ln(2)} - 2(n+1) + 2 = (n+1) \frac{\ln(n+1)}{\ln(2)} - 2n.$$

Il s'agit donc de montrer par récurrence forte la propriété $\mathcal{Q}(n)$: « $\alpha_n \geq g(n+1)$ ».

Initialisation : $\alpha_1 = 0$ et $g(2) = 0$, donc $\mathcal{Q}(1)$ est vraie. $\alpha_2 = 1$ et $g(3) = 3 \frac{\ln(3)}{\ln(2)} - 4 \leq 1$ (vu en 5) b) ii)), donc $\mathcal{Q}(2)$ est vraie.

Hérédité : soit $n \geq 3$ tel que $\mathcal{Q}(m)$ soit vraie pour tout $m \in \{1, \dots, n-1\}$. D'après 5) a), il suffit de montrer que chacun des nombres dont α_n majore le minimum est supérieur ou égal à $g(n+1)$.

Par $\mathcal{Q}(n-1)$ puis 5) b) ii) :

$$n-1+\alpha_{n-1} \geq n-1+g(n) \geq g(n+1)$$

Pour $2 \leq i \leq n-1$, on a $1 \leq i-1 \leq n-2$ et $1 \leq n-i \leq n-2$, donc par $\mathcal{Q}(i-1)$, $\mathcal{Q}(n-i)$ puis 5) b) i) :

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

Or

$$\frac{\ln\left(\frac{n+1}{2}\right)}{\ln(2)} = \frac{\ln(n+1)}{\ln(2)} - 1,$$

donc :

$$2g\left(\frac{n+1}{2}\right) = (n+1)\left(\frac{\ln(n+1)}{\ln(2)} - 1\right) - 2(n+1) + 4 = (n+1)\frac{\ln(n+1)}{\ln(2)} - 3n + 1$$

et

$$n-1+2g\left(\frac{n+1}{2}\right) = (n+1)\frac{\ln(n+1)}{\ln(2)} - 2n = g(n+1).$$

Tous les nombres considérés sont donc supérieurs ou égaux à $g(n+1)$, leur minimum aussi, et $\alpha_n \geq g(n+1)$: $\mathcal{Q}(n)$ est vraie.

Conclusion

pour tout $n \geq 1$, $\alpha_n \geq (n+1) \frac{\ln(n+1)}{\ln(2)} - 2n$. Avec 3), presque sûrement $(n+1) \frac{\ln(n+1)}{\ln(2)} - 2n \leq X_n \leq \frac{n(n-1)}{2}$.

Partie III : Étude d'un algorithme de tri

A. 1) L'univers est l'ensemble des $n!$ permutations de E , muni de la probabilité uniforme ; on dénombre les permutations réalisant $[T_1 = e_j]$.

Soit $j \in \{1, \dots, n\}$. Une permutation σ vérifie $\sigma(e_1) = e_j$ si et seulement si sa restriction à $E \setminus \{e_1\}$ est une bijection de $E \setminus \{e_1\}$ sur $E \setminus \{e_j\}$; il y a $(n-1)!$ telles bijections. Donc :

$$P([T_1 = e_j]) = \frac{(n-1)!}{n!} = \frac{1}{n}$$

Conclusion

T_1 suit la loi uniforme sur E .

2) a) Une liste $(i_1, \dots, i_k)$ strictement croissante d'éléments de $\{2, \dots, n\}$ est entièrement déterminée par l'ensemble $\{i_1, \dots, i_k\}$ de ses termes, et réciproquement toute partie à k éléments de $\{2, \dots, n\}$, rangée dans l'ordre croissant, fournit une telle liste. On a donc une bijection avec les parties à k éléments d'un ensemble à $n-1$ éléments.

Conclusion

il y a $\binom{n-1}{k}$ telles listes.

b) Notons $A_{(\alpha)}$ l'événement $\bigcap_{j=1}^k [T'_j = \alpha_j] \cap [T_1 = e_{k+1}]$.

Préliminaire. Si $\sigma(e_1) = e_{k+1}$, les éléments de E inférieurs à e_{k+1} sont exactement $e_1, \dots, e_k$; comme σ est bijective, ils figurent tous, une fois chacun, dans la liste

$(\sigma(e_1), \dots, \sigma(e_n))$, à des positions de $\{2, \dots, n\}$. Ainsi $(T'_1(\sigma), \dots, T'_k(\sigma))$ est la liste de $e_1, \dots, e_k$ rangés dans l'ordre de leurs positions. D'autre part, $(\alpha_1, \dots, \alpha_k)$ étant formée de k éléments distincts de l'ensemble $\{e_1, \dots, e_k\}$ à k éléments, on a $\{\alpha_1, \dots, \alpha_k\} = \{e_1, \dots, e_k\}$.

Méthode 1 — Dénombrement.

Une permutation σ appartient à $A_{(\alpha)}$ si et seulement si $\sigma(e_1) = e_{k+1}$ et, en notant $i_1 < \dots < i_k$ les positions occupées par les éléments inférieurs à e_{k+1} , $\sigma(e_{i_j}) = \alpha_j$ pour tout j . On construit donc un élément de $A_{(\alpha)}$, de façon unique, par les choix successifs suivants :

- choix des positions $2 \leq i_1 < \dots < i_k \leq n$: $\binom{n-1}{k}$ possibilités d'après a) ;
- placement de α_j en position i_j pour tout j : une seule possibilité ;
- placement des $n - 1 - k$ éléments $e_{k+2}, \dots, e_n$ dans les $n - 1 - k$ positions restantes : $(n - 1 - k)!$ possibilités.

Donc $\text{Card}(A_{(\alpha)}) = \binom{n-1}{k} (n - 1 - k)! = \frac{(n-1)!}{k!}$ et, la probabilité étant uniforme :

$$P(A_{(\alpha)}) = \frac{(n-1)!}{k! n!} = \frac{1}{n k!}$$

Méthode 2 — Argument de symétrie.

Soit τ une permutation de $\{e_1, \dots, e_k\}$, prolongée en une permutation $\tilde{\tau}$ de E fixant $e_{k+1}, \dots, e_n$. L'application $\Phi : \sigma \mapsto \tilde{\tau} \circ \sigma$ est une bijection de l'ensemble des permutations de E sur lui-même (de réciproque $\sigma \mapsto \tilde{\tau}^{-1} \circ \sigma$). Si $\sigma(e_1) = e_{k+1}$, alors $\Phi(\sigma)(e_1) = e_{k+1}$, les éléments inférieurs à e_{k+1} occupent dans $\Phi(\sigma)$ les mêmes positions que dans σ , et

l'élément situé à chacune de ces positions est transformé par τ . Donc Φ envoie $A_{(\alpha)}$ dans $A_{(\tau(\alpha_1), \dots, \tau(\alpha_k))}$, et $\tilde{\tau}^{-1}$ fournit l'application réciproque : ces deux événements ont même cardinal.

Pour deux listes β, β' formées des éléments de $\{e_1, \dots, e_k\}$ rangés dans des ordres quelconques, il existe une permutation τ telle que $\tau(\beta_j) = \beta'_j$ pour tout j . Les $k!$ événements $A_{(\beta)}$ ont donc la même probabilité. D'après le préliminaire, ils forment une partition de $[T_1 = e_{k+1}]$, de probabilité $\frac{1}{n}$ d'après 1). Donc $k! P(A_{(\alpha)}) = \frac{1}{n}$.

La méthode 1 est un dénombrement direct ; la méthode 2 exploite l'invariance de la probabilité uniforme par renommage des éléments et évite tout calcul de cardinal. Cette idée de symétrie resservira en III.B.2) a).

Conclusion

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

- c) D'après 1), $P([T_1 = e_{k+1}]) = \frac{1}{n} \neq 0$, donc la probabilité conditionnelle est définie et, d'après b) :

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

Il y a exactement $k!$ listes d'éléments distincts de $\{e_1, \dots, e_k\}$ et ces $k!$ probabilités conditionnelles ont pour somme 1.

Conclusion

sachant $[T_1 = e_{k+1}]$, le vecteur $(T'_1, \dots, T'_k)$ suit la loi uniforme sur l'ensemble des $k!$ listes formées des éléments $e_1, \dots, e_k$ rangés dans un ordre quelconque.

- B. Code adapté en Python. Les déclarations Pascal et les procédures Placer et Tri de l'énoncé original ont été remplacées par une liste Python T indexée de 0 à $n - 1$, une

fonction `placer` qui modifie la liste en place et renvoie l'indice `pl`, et une fonction récursive `tri`. Tous les indices et exemples ont été décalés de 1 ; le comportement mathématique (sous-tableaux traités, nombre de comparaisons, position finale du pivot) est inchangé.

Pour pouvoir exécuter et vérifier les réponses, voici une programmation possible de `placer`, non demandée par l'énoncé et conforme à sa description : chacun des `fin - deb` éléments autres que le pivot `T[deb]` est comparé une seule fois au pivot, puis rangé dans la liste petits ou grands en conservant l'ordre d'apparition.

Python

```
def placer(T, deb, fin):
    if fin <= deb:
        return deb

    pivot = T[deb]
    petits = []
    grands = []

    for k in range(deb + 1, fin + 1):
        if T[k] < pivot:           # une comparaison pour chaque k
            petits = petits + [T[k]]
        else:
            grands = grands + [T[k]]

    L = petits + [pivot] + grands
    for k in range(deb, fin + 1):
        T[k] = L[k - deb]

    return deb + len(petits)


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)
```

Ce code a été exécuté : sur $T = [12, 3, 8, 10, 6, 4, 5]$, $pl = \text{placer}(T, 2, 5)$ renvoie 4 et transforme T en $[12, 3, 6, 4, 8, 10, 5]$, et $pl = \text{placer}(T, 2, 3)$ renvoie 2 sans modifier T , conformément aux exemples de l'énoncé. La boucle `for` parcourt les indices $deb+1$ à fin inclus (la borne supérieure de range est exclue, d'où $fin + 1$) ; la valeur renvoyée $deb + \text{len}(\text{petits})$ est bien la nouvelle position du pivot.

- 1) a) Question adaptée au programme actuel (Code adapté en Python : i varie de 0 à $n - 1$ au lieu de 1 à n).

Lors de l'instruction $\text{tri}(T, i, i)$, on a $fin = deb = i$, donc le test $fin > deb$ est faux : aucune instruction n'est exécutée, en particulier `placer` n'est pas appelée.

Conclusion

l'instruction $\text{tri}(T, i, i)$ laisse la liste T inchangée (et n'effectue aucune comparaison).

- b) Question adaptée au programme actuel (Code adapté en Python : l'instruction $\text{Tri}(T, 1, 5)$ de l'énoncé original devient $\text{tri}(T, 0, 4)$).

On suit l'exécution en se rappelant que chaque appel de `tri` possède sa propre variable locale pl et que la liste T est modifiée en place. Initialement $T = [2, 9, 6, 1, 5]$.

Étape	Appel	Effet
1	$\text{tri}(T, 0, 4)$	$4 > 0$: appel de $\text{placer}(T, 0, 4)$
2	$\text{placer}(T, 0, 4)$	pivot 2 ; plus petit : 1 ; plus grands : 9, 6, 5 ; $T = [1, 2, 9, 6, 5]$, $pl = 1$
3	$\text{tri}(T, 0, 0)$	appelé car $pl = 1 > 0$; ne fait rien
4	$\text{tri}(T, 2, 4)$	appelé car $pl = 1 < 4$; $4 > 2$: appel de $\text{placer}(T, 2, 4)$

Étape	Appel	Effet
5	<code>placer(T, 2, 4)</code>	pivot 9 ; plus petits : 6, 5 ; $T = [1, 2, 6, 5, 9]$, $pl = 4$
6	<code>tri(T, 2, 3)</code>	appelé car $pl = 4 > 2$; $3 > 2$: appel de <code>placer(T, 2, 3)</code>
7	<code>placer(T, 2, 3)</code>	pivot 6 ; plus petit : 5 ; $T = [1, 2, 5, 6, 9]$, $pl = 3$
8	<code>tri(T, 2, 2)</code>	appelé car $pl = 3 > 2$; ne fait rien

Dans l'appel `tri(T, 2, 3)`, on a $pl = 3 = \text{fin}$, donc `tri(T, 4, 3)` n'est pas appelé ; de même, dans l'appel `tri(T, 2, 4)`, $pl = 4 = \text{fin}$, donc il n'y a pas de second appel récursif. L'exécution se termine avec $T = [1, 2, 5, 6, 9]$, après $4 + 2 + 1 = 7$ comparaisons.

Cette trace a été vérifiée par exécution de la version suivante de `tri`, qui affiche les appels :

Python

```
def tri_trace(T, deb, fin):
    print("appel tri(T, " + str(deb) + ", " + str(fin) + ")")
    if fin > deb:
        pl = placer(T, deb, fin)
        print("  placer(T, " + str(deb) + ", " + str(fin) + ") : pl = " + str(pl) + ", T = " + str(T))
        if pl > deb:
            tri_trace(T, deb, pl - 1)
        if pl < fin:
            tri_trace(T, pl + 1, fin)

T = [2, 9, 6, 1, 5]
tri_trace(T, 0, 4)
```

La sortie obtenue reproduit exactement, ligne à ligne, les appels et les affectations du tableau ci-dessus.

Conclusion

les appels successifs sont $\text{tri}(T, 0, 4)$, $\text{placer}(T, 0, 4)$, $\text{tri}(T, 0, 0)$, $\text{tri}(T, 2, 4)$, $\text{placer}(T, 2, 4)$, $\text{tri}(T, 2, 3)$, $\text{placer}(T, 2, 3)$, $\text{tri}(T, 2, 2)$, et la liste finale est $T = [1, 2, 5, 6, 9]$.

- c) Effet : $\text{tri}(T, \text{deb}, \text{fin})$ range par ordre croissant les éléments du sous-tableau $[T[\text{deb}], \dots, T[\text{fin}]]$, sans modifier les autres éléments de T . En particulier $\text{tri}(T, 0, n-1)$ trie la liste T .

Principe (tri « rapide », de type « diviser pour régner ») : l'élément $T[\text{deb}]$ sert de pivot ; placer le met à sa place définitive, avec à sa gauche les éléments plus petits et à sa droite les éléments plus grands ; il reste alors à trier, par deux appels récursifs, la partie gauche $[T[\text{deb}], \dots, T[\text{pl}-1]]$ et la partie droite $[T[\text{pl}+1], \dots, T[\text{fin}]]$.

Justification de l'arrêt et de la correction, par récurrence forte sur la longueur $m = \text{fin} - \text{deb} + 1$ du sous-tableau. Propriété $\mathcal{R}(m)$: pour tous deb, fin avec $\text{fin} - \text{deb} + 1 = m$, l'appel $\text{tri}(T, \text{deb}, \text{fin})$ se termine, ne modifie que $[T[\text{deb}], \dots, T[\text{fin}]]$, en permutant ses éléments, et le laisse trié.

Si $m \leq 1$, l'appel ne fait rien et un sous-tableau d'au plus un élément est trié.

Si $m \geq 2$, on suppose $\mathcal{R}(m')$ vraie pour tout $m' < m$. L'appel de placer se termine, ne fait que permuter les éléments du sous-tableau et renvoie $\text{pl} \in \{\text{deb}, \dots, \text{fin}\}$. Les éventuels appels récursifs portent sur des sous-tableaux de longueurs $\text{pl} - \text{deb} \leq m - 1$ et $\text{fin} - \text{pl} \leq m - 1$, strictement inférieures à m : ils se terminent, chacun ne modifie que sa partie et la trie. Comme ces parties sont disjointes et que chacune ne fait que permuter ses propres éléments, la partie gauche contient toujours exactement les éléments inférieurs au pivot et la partie droite ceux qui lui sont supérieurs. À la fin, la partie gauche est triée, puis vient le pivot, puis la partie droite triée : le sous-tableau est trié.

Conclusion

la fonction tri trie le sous-tableau $[T[\text{deb}], \dots, T[\text{fin}]]$ par ordre croissant. L'algorithme s'arrête car chaque appel récursif porte sur un sous-tableau strictement plus court que celui de l'appel qui le lance (le pivot en est exclu), et un sous-tableau de longueur au plus 1 ne provoque aucun appel.

- 2) Question adaptée au programme actuel (Code adapté en Python : $T[0] = \sigma(e_1), \dots, T[n-1] = \sigma(e_n)$ et l'instruction étudiée est $\text{tri}(T, 0, n-1)$).

Remarque : l'univers étant fini et muni de la probabilité uniforme, chaque permutation σ a une probabilité $\frac{1}{n!} > 0$. Une propriété presque sûre des X_n est donc ici vraie pour toute permutation σ .

- a) Lemme préliminaire. Pour une liste $L = (a_1, \dots, a_m)$ de nombres distincts placée dans un sous-tableau, notons $C(L)$ le nombre de comparaisons effectuées par les appels de placer lors de l'exécution de tri sur ce sous-tableau, avec $C(L) = 0$ si $m \leq 1$ (et pour la liste vide). Ce nombre ne dépend que de l'ordre relatif des a_r , c'est-à-dire des réponses aux tests $a_r < a_s$, et pas de la position du sous-tableau dans T . En effet, placer effectue toujours exactement $m - 1$ comparaisons, la valeur pl renvoyée et l'ordre relatif des éléments du sous-tableau obtenu ne dépendent que de l'ordre relatif des éléments de départ, et les appels récursifs portent sur des sous-tableaux plus courts ; une récurrence forte sur m conclut.

Par conséquent, si F est un ensemble de $m \geq 1$ réels et si L est une liste aléatoire uniforme parmi les $m!$ façons de ranger les éléments de F , alors $C(L)$ a la loi de X_m . En effet, la bijection croissante de F sur un ensemble $\{e'_1 < \dots < e'_m\}$ servant à définir X_m transforme une liste uniforme d'éléments de F en une liste uniforme d'éléments de $\{e'_1, \dots, e'_m\}$ ayant même ordre relatif, donc même nombre de comparaisons. En particulier la loi de X_m ne dépend pas de l'ensemble E choisi.

Définition de I_n . Posons $I_n(\sigma) = i$ lorsque $T_1(\sigma) = e_i$, c'est-à-dire que I_n est le rang du pivot initial. D'après A.1), I_n suit la loi uniforme sur $\{1, \dots, n\}$.

Cas $n = 1$: $\text{tri}(T, 0, 0)$ ne fait rien d'après 1) a), donc $X_1 = 0$.

Soit $n \geq 2$ et $i \in \{1, \dots, n\}$. Sur l'événement $[I_n = i]$, le premier appel $\text{placer}(T, 0, n-1)$

effectue $n - 1$ comparaisons avec le pivot e_i . Comme les éléments glissent sans permutation entre eux, il transforme T en :

$$[T'_1, \dots, T'_{i-1}, e_i, T''_1, \dots, T''_{n-i}]$$

où $L = (T'_1, \dots, T'_{i-1})$ est la liste des éléments inférieurs à e_i dans leur ordre d'apparition (notations de A.2)) et $R = (T''_1, \dots, T''_{n-i})$ celle des éléments supérieurs à e_i dans leur ordre d'apparition. La valeur renvoyée est $pl = i - 1$. Les appels récursifs $\text{tri}(T, 0, i-2)$ (si $i \geq 2$) et $\text{tri}(T, i, n-1)$ (si $i \leq n - 1$) trient deux sous-tableaux disjoints, chacun sans modifier l'autre (question 1) c)). Donc, sur $[I_n = i]$:

$$X_n = n - 1 + C(L) + C(R)$$

Loi conditionnelle de (L, R) sachant $[I_n = i]$. Soit β une liste formée des éléments de $\{e_1, \dots, e_{i-1}\}$ dans un ordre quelconque et γ une liste formée des éléments de $\{e_{i+1}, \dots, e_n\}$ dans un ordre quelconque. Comme en A.2) b), une permutation réalisant $[T_1 = e_i] \cap [L = \beta] \cap [R = \gamma]$ est entièrement déterminée par le choix des $i - 1$ positions, parmi les $n - 1$ positions $2, \dots, n$, occupées par les éléments inférieurs à e_i ; il y en a $\binom{n-1}{i-1}$. Donc :

$$P([L = \beta] \cap [R = \gamma] / [T_1 = e_i]) = \frac{n}{n!} \binom{n-1}{i-1} = \frac{1}{(i-1)!(n-i)!}$$

En sommant sur les $(n - i)!$ listes γ , on retrouve $P([L = \beta] / [T_1 = e_i]) = \frac{1}{(i-1)!}$, ce qui est le résultat de A.2) c) avec $k = i - 1$; en sommant sur les $(i - 1)!$ listes β , on obtient de même $P([R = \gamma] / [T_1 = e_i]) = \frac{1}{(n-i)!}$. La loi conjointe est donc le produit des lois marginales : pour la probabilité conditionnelle sachant $[I_n = i]$, les listes L et R sont indépendantes, L est uniforme parmi les rangements de $\{e_1, \dots, e_{i-1}\}$ et R uniforme parmi les rangements de $\{e_{i+1}, \dots, e_n\}$.

Vérification des hypothèses de II.B.

- $X_1 = 0$.
- Si $i = 1$: L est vide, donc $X_n = n - 1 + C(R)$ avec R uniforme parmi les rangements d'un ensemble à $n - 1$ éléments ; d'après le lemme, la loi conditionnelle de X_n sachant $[I_n = 1]$ est celle de $n - 1 + X_{n-1}$. Si $i = n$: R est vide et $L = (T'_1, \dots, T'_{n-1})$ est uniforme d'après A.2) c) avec $k = n - 1$; la loi conditionnelle de X_n sachant $[I_n = n]$ est encore celle de $n - 1 + X_{n-1}$.
- Si $n \geq 3$ et $2 \leq i \leq n - 1$: sachant $[I_n = i]$, $C(L)$ et $C(R)$ sont indépendantes (fonctions de variables indépendantes), de lois respectives celles de X_{i-1} et X_{n-i} d'après le lemme. Or la loi de la somme de deux variables indépendantes ne dépend que de leurs lois : pour tout entier k ,

$$P([Z + T = k]) = \sum_j P([Z = j]) P([T = k - j]).$$

Donc la loi conditionnelle de X_n sachant $[I_n = i]$ est celle de $n - 1 + Z_{n,i} + T_{n,i}$ avec $Z_{n,i}$, $T_{n,i}$ indépendantes de mêmes lois que X_{i-1} et X_{n-i} .

Remarque : en II.B, toutes les variables sont définies sur un même espace, alors qu'ici X_n est définie sur l'ensemble des permutations d'un ensemble à n éléments. Les résultats de la partie II utilisés (questions 4), 5) et 6)) ne font intervenir que les lois des X_n et leurs lois conditionnelles sachant les événements $[I_n = i]$, qui sont exactement celles qu'on vient d'établir : ils s'appliquent.

Conséquence. D'après II.4) b) :

$$E(X_n) = U_n = 2(n+1) \sum_{k=2}^n \frac{1}{k} - 2n + 2$$

Conclusion

la suite (X_n) vérifie les hypothèses de II.B et $E(X_n) \sim 2n \ln n$ quand n tend vers l'infini.

Illustration numérique (non demandée, ne constituant pas une preuve). Le programme suivant, exécuté, estime $E(X_{100})$ par la moyenne empirique de 10000 simulations. La fonction `permutation_aleatoire` mélange la liste $[1, \dots, n]$ par la méthode de Fisher-Yates (échange de $T[k]$ avec $T[j]$ pour j entier uniforme entre 0 et k , k décroissant de $n-1$ à 1); on admet qu'elle fournit une permutation uniforme. Rappel : `rd.randint(a, b)` renvoie un entier uniforme entre a et $b-1$.

Python

```
import numpy as np
import numpy.random as rd

def tri_compte(T, deb, fin):
    # trie T[deb..fin] comme tri et renvoie le nombre de comparaisons
    c = 0
    if fin > deb:
        pl = placer(T, deb, fin)
        c = fin - deb
        if pl > deb:
            c = c + tri_compte(T, deb, pl - 1)
        if pl < fin:
            c = c + tri_compte(T, pl + 1, fin)
    return c

def permutation_aleatoire(n):
    T = list(range(1, n + 1))
    for k in range(n - 1, 0, -1):
        j = rd.randint(0, k + 1)      # entier uniforme entre 0 et k
        x = T[k]
```

```

        T[k] = T[j]

        T[j] = x

    return T

def U(n):
    h = 0

    for k in range(2, n + 1):
        h = h + 1 / k

    return 2 * (n + 1) * h - 2 * n + 2

n = 100
N = 10000
S = np.zeros(N)
for r in range(N):
    T = permutation_aleatoire(n)
    S[r] = tri_compte(T, 0, n - 1)
print(np.mean(S), U(n), 2 * n * np.log(n))

```

Une exécution a donné une moyenne empirique d'environ 648,1, pour une valeur exacte $U_{100} \approx 647,85$, tandis que $2n \ln n \approx 921$: l'équivalent $2n \ln n$ n'est atteint que lentement, car $U_n - 2n \ln n$ est de l'ordre de n .

b) Pour $E = \{1, \dots, n\}$, on propose la liste déjà triée $T = [1, 2, \dots, n]$.

Montrons par récurrence sur $m \geq 1$ qu'un sous-tableau de longueur m déjà trié par ordre croissant demande $\frac{m(m-1)}{2}$ comparaisons. Pour $m = 1$, aucune comparaison n'est faite. Pour $m \geq 2$: le pivot est le plus petit élément ; placer effectue $m - 1$ comparaisons, tous les autres éléments sont plus grands et glissent à droite sans permutation, donc le sous-tableau est inchangé et $pl = deb$. Il n'y a pas d'appel à gauche, et l'appel à droite porte sur un sous-tableau trié de longueur $m - 1$, qui demande $\frac{(m-1)(m-2)}{2}$ comparaisons par hypothèse de récurrence. Au total :

$$m - 1 + \frac{(m-1)(m-2)}{2} = \frac{m(m-1)}{2}.$$

Commentaire : à chaque étape le pivot est un élément extrême de son sous-tableau, donc le découpage est le plus déséquilibré possible (une partie vide, l'autre de longueur $m - 1$) ; on obtient $(n - 1) + (n - 2) + \dots + 1$ comparaisons. D'après II.3), c'est le nombre maximal possible de comparaisons. La liste $[n, n-1, \dots, 1]$ convient aussi : le pivot est alors toujours le plus grand élément et la partie restante $[n-1, \dots, 1]$ a la même forme.

Conclusion

la liste $[1, 2, \dots, n]$ nécessite $\frac{n(n-1)}{2}$ comparaisons pour être triée.

c) Pour $E = \{1, \dots, 7\}$, on propose $T = [4, 2, 1, 3, 6, 5, 7]$, dans laquelle chaque pivot est l'élément médian de son sous-tableau.

- $\text{placer}(T, 0, 6)$: 6 comparaisons, pivot 4 ; T devient $[2, 1, 3, 4, 6, 5, 7]$ et $pl = 3$.
- $\text{tri}(T, 0, 2)$: $\text{placer}(T, 0, 2)$ effectue 2 comparaisons, pivot 2 ; T devient $[1, 2, 3, 4, 6, 5, 7]$, $pl = 1$; les appels $\text{tri}(T, 0, 0)$ et $\text{tri}(T, 2, 2)$ ne font rien.
- $\text{tri}(T, 4, 6)$: $\text{placer}(T, 4, 6)$ effectue 2 comparaisons, pivot 6 ; T devient $[1, 2, 3, 4, 5, 6, 7]$, $pl = 5$; les appels $\text{tri}(T, 4, 4)$ et $\text{tri}(T, 6, 6)$ ne font rien.

Total : $6 + 2 + 2 = 10$ comparaisons (valeur confirmée par exécution de `tri_compte`).

Commentaire : le pivot coupe toujours son sous-tableau en deux parties de même longueur (3 et 3, puis 1 et 1), ce qui est le cas le plus favorable. D'ailleurs, d'après II.6), pour toute permutation $X_7 \geq \alpha_7 \geq g(8) = 8 \cdot 3 - 16 + 2 = 10$: on ne peut pas trier une liste de 7 éléments avec moins de 10 comparaisons par cet algorithme.

Conclusion

la liste $[4, 2, 1, 3, 6, 5, 7]$ nécessite 10 comparaisons pour être triée.

d) On généralise l'exemple de c) en choisissant à chaque étape le pivot médian, ce qui suppose des longueurs de la forme $2^p - 1$. On montre que les entiers $n = 2^p - 1$, $p \geq 1$, conviennent.

Calcul de $g(n+1)$ pour $n = 2^p - 1$: $g(2^p) = 2^p \cdot p - 2^{p+1} + 2 = (p-2)2^p + 2$.

Construction. Pour $p \geq 1$ et a entier, on définit la liste $B_p(a)$, formée des $2^p - 1$ entiers consécutifs $a, a+1, \dots, a+2^p-2$, par $B_1(a) = [a]$ et, pour $p \geq 1$, en notant $m = a + 2^p - 1$:

$B_{p+1}(a) = [m]$ suivie de $B_p(a)$, suivie de $B_p(m+1)$

La liste $B_p(a)$ contient les entiers de a à $m-1 = a+2^p-2$ et la liste $B_p(m+1)$ ceux de $m+1$ à $a+2^{p+1}-2$; $B_{p+1}(a)$ contient donc bien les $2^{p+1}-1$ entiers de a à $a+2^{p+1}-2$, et m en est l'élément médian. Par exemple $B_2(1) = [2, 1, 3]$ et $B_3(1) = [4, 2, 1, 3, 6, 5, 7]$, la liste de c).

Nombre de comparaisons. Montrons par récurrence sur $p \geq 1$ que

$C(B_p(a)) = (p-2)2^p + 2$ pour tout a . Pour $p = 1$: $C([a]) = 0 = (1-2) \cdot 2 + 2$. Si c'est vrai au rang p : placer sur $B_{p+1}(a)$, de longueur $2^{p+1}-1$, effectue $2^{p+1}-2$ comparaisons, avec pivot m ; les éléments plus petits que m sont ceux de $B_p(a)$, qui apparaissent avant ceux de $B_p(m+1)$, et ils glissent sans permutation, donc le sous-tableau devient $B_p(a)$, puis m , puis $B_p(m+1)$. Les deux appels récursifs trient $B_p(a)$ et $B_p(m+1)$, ce qui, d'après le lemme de a) et l'hypothèse de récurrence, demande $(p-2)2^p + 2$ comparaisons chacun. Au total :

$$\begin{aligned} 2^{p+1} - 2 + 2((p-2)2^p + 2) &= 2^{p+1} + (p-2)2^{p+1} + 2 = (p-1)2^{p+1} + 2 \\ &= ((p+1)-2)2^{p+1} + 2 \end{aligned}$$

Ainsi, pour $n = 2^p - 1$, la liste $B_p(1)$ d'éléments de $\{1, \dots, n\}$ nécessite exactement $g(n+1)$ comparaisons. D'après II.6), c'est le minimum possible, donc $\alpha_{2^p-1} = g(2^p)$.

Vérification par exécution :

Python

```
def liste_equilibree(a, p):
    # liste B_p(a) contenant les entiers a, a+1, ..., a + 2**p - 2
    if p == 1:
        return [a]
    m = a + 2 ** (p - 1) - 1
```

```
return [m] + liste_equilibree(a, p - 1) + liste_equilibree(m + 1, p - 1)
```

```
for p in range(1, 7):
    n = 2 ** p - 1
    T = liste_equilibree(1, p)
    print(n, tri_compte(T, 0, n - 1), (p - 2) * 2 ** p + 2)
```

Les couples obtenus (nombre de comparaisons, valeur de $g(n+1)$) sont $(0, 0)$, $(2, 2)$, $(10, 10)$, $(34, 34)$, $(98, 98)$, $(258, 258)$ pour $n = 1, 3, 7, 15, 31, 63$. Ici l'appel `liste_equilibree(a, p)` construit $B_p(a)$ à partir de B_{p-1} , avec le médian $m = a + 2^{p-1} - 1$.

Complément : ce sont les seuls entiers possibles. Si $n+1$ n'est pas une puissance de 2, alors $\frac{\ln(n+1)}{\ln(2)}$ est irrationnel : si $\frac{\ln(n+1)}{\ln(2)} = \frac{a}{b}$ avec a, b entiers naturels non nuls, alors $(n+1)^b = 2^a$, donc tout facteur premier de $n+1$ divise 2^a et vaut 2, et $n+1$ serait une puissance de 2. Dans ce cas $g(n+1) = (n+1)\frac{\ln(n+1)}{\ln(2)} - 2n$ n'est pas un entier, alors qu'un nombre de comparaisons est entier.

Conclusion

pour les entiers $n = 2^p - 1$ ($p \geq 1$), soit $1, 3, 7, 15, \dots$, la liste $B_p(1)$ nécessite exactement $g(n+1) = (p-2)2^p + 2$ comparaisons, et ce sont les seuls entiers n pour lesquels une telle liste existe.

Relevé des adaptations

- III.B (préambule) : Code adapté en Python. Les déclarations Pascal (`const n, type tableau = array[1..n] of integer`) sont remplacées par une liste Python `T` de longueur n indexée de 0 à $n-1$; la procédure `Placer(var T, deb, fin, var pl)` devient la fonction `placer(T, deb, fin)`, qui modifie la liste en place et renvoie `pl` ; la procédure `Tri` devient la fonction récursive `tri`, de même structure. Justification : seul Python est au programme ; le comportement mathématique est conservé.

- III.B (préambule) : bornes $1 \leq \text{deb} < \text{fin} \leq n$ remplacées par $0 \leq \text{deb} < \text{fin} \leq n - 1$ et exemples réindexés : $T[3]$ devient $T[2]$ dans l'exemple $[3, 1, 7, 5]$; $\text{Placer}(T, 3, 6, \text{pl})$ devient $\text{placer}(T, 2, 5)$ avec $\text{pl} = 4$ au lieu de 5 ; $\text{Placer}(T, 3, 4, \text{pl})$ devient $\text{placer}(T, 2, 3)$ avec $\text{pl} = 2$ au lieu de 3. Justification : indexation des listes Python à partir de 0.
- III.B (préambule) : pour $\text{fin} \leq \text{deb}$, « ne fait rien » est précisé en « ne modifie pas T et renvoie deb ». Justification : une fonction Python renvoie une valeur ; sans incidence, car tri n'appelle placer que si $\text{fin} > \text{deb}$.
- III.B (préambule) : le type tableau d'entiers est remplacé par une liste de nombres distincts. Justification : en III.B.2), la liste reçoit les réels $\sigma(e_i)$, ce que le type integer de l'énoncé original ne permettait pas ; le traitement mathématique est inchangé.
- III.B.1) a) : Question adaptée au programme actuel ; l'entier i varie de 0 à $n - 1$ au lieu de 1 à n .
- III.B.1) b) : Question adaptée au programme actuel ; $\text{Tri}(T, 1, 5)$ devient $\text{tri}(T, 0, 4)$, et « procédures successives » devient « appels successifs des fonctions tri et placer ».
- III.B.2) : Question adaptée au programme actuel ; $T[1], \dots, T[n]$ deviennent $T[0], \dots, T[n-1]$ et $\text{Tri}(T, 1, n)$ devient $\text{tri}(T, 0, n-1)$; le mot « tableau » devient « liste » en 2) b), c), d). Justification : cohérence avec le code Python.