DS3 – Mathématiques

Mercredi 13 Novembre 2024

Durée : 3 heures

- **Aucun document ou appareil électronique n'est autorisé.**
- **Rédigez l'exercice d'informatique sur une copie séparée.**
- Le devoir comporte deux exercices de mathématiques (exercices 1 et 3), un exercice d'informatique (exercice 2) et un problème. Je vous recommande de chercher les exercices dans l'ordre proposé par le sujet.
- La qualité de la présentation et de la rédaction, la lisibilité et la précision seront prises en compte pour une part importante de la note finale.
- *Encadrez* ou *soulignez* les résultats de vos calculs et les conclusions de vos raisonnements.

Exercice 1 (Cours et TD)**.**

1. Soit $n \in \mathbb{N}$. Calculer les sommes suivantes :
   - $A = \sum_{k=0}^{n} k^2(k+2)$,
   - $B = \sum_{k=1}^{n} 3$,
   - $C = \sum_{k=1}^{n} \ln\left(\frac{k+1}{k}\right)$,
   - $D = \sum_{j=n}^{2n} j$.
2. (a) Rappeler la formule du binôme de Newton.

   (b) Calculer 12^5 à l'aide de la formule du binôme, en détaillant votre calcul.

   (c) Soit $n \in \mathbb{N}$. Calculer $S_n = \sum_{k=1}^{n} \binom{n}{k} 3^k \times (-1)^{n-k}$.
3. Soit (u_n) la suite définie par $u_0 = 1$, $u_1 = 0$ et pour tout $n \in \mathbb{N}$,
$$u_{n+2} = 4u_{n+1} - 3u_n.$$
   (a) Déterminer l'expression de u_n en fonction de $n \in \mathbb{N}$.

   (b) Exprimer
$$\sum_{k=0}^{n} u_k$$
   en fonction de $n \in \mathbb{N}$.

Exercice 2 (Informatique - langage Python - à rédiger sur une feuille séparée)**.**

1. Écrire une fonction Python `factorielle` qui prend en argument un entier n et qui renvoie $n!$, la factorielle de n.
2. Combien de multiplications sont effectuées par la machine lors de l'appel `factorielle(5)` ? Lors de l'appel `factorielle(n)` ?
3. Rappeler la définition, pour k et $n \in \mathbb{N}$, du coefficient binomial $\binom{n}{k}$.
4. Écrire une fonction Python `binom` qui prend en argument deux entiers n et k et qui renvoie $\binom{n}{k}$ *(indication : on utilisera la fonction `factorielle` écrite à la question 1.)*
5. Calculez $\binom{10}{3}$. On veillera à simplifier au maximum avant de se lancer dans les calculs. Combien de multiplications et de divisions avez-vous effectué pour parvenir au résultat ?
6. Combien de multiplications et de divisions sont effectuées par la machine lors de l'appel `binom(10,3)` ? L'implémentation de la fonction `binom` vous paraît-elle raisonnable ? Pourquoi ?
7. Montrer que pour tout n et pour tout $k \in \mathbb{N}$,
$$\binom{n}{k} = \begin{cases} \dfrac{n \times (n-1) \times \cdots \times (n-k+1)}{k!} & \text{si } k \leq n, \\ 0 \text{ sinon.} \end{cases} \tag{1}$$

8. En déduire l'implémentation d'une fonction `binom2` qui prend en argument deux entiers n et k et qui renvoie $\binom{n}{k}$ en utilisant la relation (1).
9. Combien de multiplications et de divisions sont effectuées par la machine lors de l'appel `binom2(10,3)` ? Comparez les deux implémentations, `binom` et `binom2` : laquelle est la plus efficace ?

Problème (Dynamique de populations : étude de quelques modèles proies-prédateurs.)**.** La dynamique des populations consiste à étudier l'évolution du nombre d'individus d'une population au cours du temps. Le cadre dit discret consiste à supposer que le temps est un entier positif n (correspondant par exemple à un nombre de jours ou d'années). L'évolution d'une population X est alors représentée par une suite $(x_n)_{n\in\mathbb{N}}$ où x_n est le nombre d'individus au temps n.

Dans ce problème, on s'intéresse à l'évolution de deux populations : une colonie de Drosophilia suzukii (communément appelé moucheron asiatique), notée X, et une colonie de perce-oreilles, notée Y. On dispose donc de deux suites $(x_n)_{n\in\mathbb{N}}$ et $\left(y_n\right)_{n\in\mathbb{N}}$ correspondant aux évolutions de ces deux populations au cours du temps.

On cherche tout particulièrement à déterminer si la population X peut ou non survivre en présence de la population Y. On dira que la population X ne peut pas survivre lorsqu'on est face à un des deux cas suivants :

- la population s'éteint "en temps long" c'est-à-dire : $\lim_{n\to+\infty} x_n = 0$, ou
- la population s'éteint "en temps fini" c'est-à-dire : il existe $N \in \mathbb{N}$ tel que $x_N \leq 0$.

1. Dans un premier temps, on suppose que les populations X et Y n'interagissent pas, et que ces espèces ont un taux de reproduction identique $a > 0$. Cela signifie que :

$$\forall\, n \in \mathbb{N}, \begin{cases} x_{n+1} = a x_n \\ y_{n+1} = a y_n \end{cases}$$

   (a) Déterminer l'expression de x_n et de y_n en fonction de x_0, y_0, a et n.

   (b) À quelle condition la population X survit-elle lorsque $n \to +\infty$?

   (c) Ce modèle de population vous semble-t-il réaliste ? Expliquer.

   Dans la suite du problème, on suppose désormais que les perce-oreilles (Y) attaquent les Drosophilia suzukii (X) : la population Y est un prédateur et la population X est sa proie. À chaque temps n, une partie des individus de la population X est tuée par la population Y. Plus précisément, chaque perce-oreille élimine un nombre $b > 0$ de Drosophilia suzukii. Ainsi, à l'étape n, comme il y a y_n perce-oreilles, ce sont $b y_n$ Drosophilia suzukii qui sont éliminées. Cela signifie donc que :

$$\forall\, n \in \mathbb{N}, \begin{cases} x_{n+1} = a x_n - b y_n, \\ y_{n+1} = a y_n. \end{cases}$$

2. (a) Écrire une fonction Python `popu` prenant en arguments x_0, y_0, a, b et n et renvoyant x_n et y_n.
   (Indication : on rappelle qu'une fonction Python peut renvoyer deux valeurs `u` *et* `v` *en utilisant la syntaxe* `return u,v`*)*

   (b) Montrer que pour tout $n \in \mathbb{N}$,

$$x_{n+2} = 2a x_{n+1} - a^2 x_n$$

   (Indication : une récurrence n'est pas nécessaire).

   (c) i. Justifier que $x_1 = a x_0 - b y_0$.

   ii. Déduire de ce qui précède que pour tout $n \in \mathbb{N}$,

$$x_n = \left(-\frac{b}{a} y_0 n + x_0\right) a^n.$$

   (d) On suppose que $a > 1$ et on rappelle que $b > 0$. Montrer que la population X disparaît au bout d'un temps fini N qu'on exprimera en fonction de a, b, x_0 et y_0.

(e) *Application numérique :* on suppose dans cette question uniquement que le temps est mesuré en jours, que $a = 2$, $b = 4$, $x_0 = 20$ individus et que $y_0 = 5$ individus. Donner la valeur du temps N au bout duquel la population de moucherons asiatiques disparaît.

On cherche à enrichir le modèle de dynamique de populations en considérant des hypothèses plus générales. On suppose désormais que les proies (X) se reproduisent plus rapidement que les prédateurs (Y). On note a' et a les taux de reproduction respectifs des populations X et Y, et on suppose donc que $a' > a > 1$. Chaque individu de Y continue de tuer $b > 0$ individus de X à chaque temps n, et les individus de X sont toujours inoffensifs pour ceux de Y.

3. (a) Écrire les expressions de x_{n+1} et y_{n+1} en fonction de x_n, y_n, a, a' et b correspondant à cette situation.

   (b) En suivant la même démarche qu'en 2.(b), montrer que pour tout $n \in \mathbb{N}$,

$$x_{n+2} - (a + a')x_{n+1} + aa'x_n = 0.$$

   en déduire l'expression de x_n en fonction de a, a', b, x_0, y_0 et n.

   (c) Montrer alors qu'il existe une valeur seuil γ, dont on donnera l'expression en fonction des autres paramètres, telle que :

   - si $x_0 > \gamma$ alors la population X survit, et
   - si $x_0 < \gamma$ alors la population X disparaît au bout d'un temps fini N qu'on explicitera.

   Interpréter ce résultat.

Exercice 3. Le but de cet exercice est de déterminer une formule simple pour la somme suivante, pour $n \in \mathbb{N}$,

$$\sum_{k=0}^{n} k2^k.$$

Pour tout entier n, on pose $u_n = \sum_{j=0}^{n} \sum_{k=j}^{n} 2^k$.

1. Calculer u_0, u_1 et u_2.
2. (a) Soient $p \leq q$ deux entiers. Calculer $\sum_{k=p}^{q} 2^k$.

   (b) En déduire que pour tout entier naturel $n \in \mathbb{N}$, $u_n = n2^{n+1} + 1$.
3. Pour tout entier n, on pose $v_n = \sum_{k=0}^{n} (k+1)2^k$.

   Montrer que pour tout entier naturel $n \in \mathbb{N}$ on a

$$v_{n+1} - v_n = u_{n+1} - u_n$$

   et en déduire que pour tout $n \in \mathbb{N}$, $u_n = v_n$.
4. Déduire de ce qui précède une expression de la somme

$$\sum_{k=0}^{n} k2^k$$

   en fonction de $n \in \mathbb{N}$.