DS n^{o} 3 de mathématiques et d'informatique

durée : 3 heures

L'usage de la calculatrice ou de documents est interdit.
Il est recommandé de lire l'ensemble de l'énoncé attentivement et patiemment.
Si vous repérez ce qui vous semble être une erreur d'énoncé, signalez-le sur votre copie et poursuivez votre composition en mentionnant les hypothèses que vous avez été amené à formuler.
Le soin de la présentation, la qualité de la rédaction, ainsi que la rigueur, la clarté et la concision des raisonnements constitueront un facteur important d'appréciation de la copie.

Problème 1 (Modélisation mathématique et informatique)
Un modèle de dynamique des populations : les populations structurées

La dynamique des populations consiste à étudier l'évolution du nombre d'individus d'une population au cours du temps. Un des premiers modèles est de supposer que le temps est un entier positif n (correspondant par exemple à un nombre d'années) et que le taux de croissance d'une population est un réel $a > 0$ fixé. Autrement dit, si x_n est le nombre d'individus au temps n, on a $\frac{x_{n+1}}{x_n} = a$. On constate alors que $(x_n)_{n\in\mathbb{N}}$ est une suite géométrique de raison a.

Par exemple, on cherche à déterminer le nombre de cellules x_n après n heures sachant qu'une cellule se divise en deux toutes les heures. En supposant qu'il y ait au temps 0 une seule cellule et que l'on soit dans des conditions optimales, on constate alors : $\forall n \in \mathbb{N}, \quad x_{n+1} = 2x_n$. Donc pour tout $n \in \mathbb{N}$, $x_n = 2^n$ et ici le taux de croissance est 2.

Mais ce modèle comporte de nombreux inconvénients, l'un d'entre eux étant qu'il suppose que tous les individus de la population étudiée se reproduisent de la même manière.

Une façon de raffiner ce modèle est alors de partager la population en plusieurs classes suivant certains facteurs (par exemple l'âge). En faisant des hypothèses sur la croissance de la population (ici linéaire), il est alors possible d'obtenir des prévisions plus précises dans le développement de la population étudiée.

A - Exemple : un modèle de développement cellulaire

Dans cette partie, on suppose qu'il existe deux types de cellules : les matures et les immatures. Si au temps n une cellule est immature, elle devient mature au temps $n + 1$. Si au temps n une cellule est mature, alors au temps $n + 1$ elle se divise en deux cellules, l'une mature et l'autre immature. En posant respectivement x_n et y_n le nombre de cellules matures et immatures au temps n, on constate alors que les suites $(x_n)_{n\in\mathbb{N}}$ et $(y_n)_{n\in\mathbb{N}}$ vérifient la récurrence suivante :

$$\forall n \in \mathbb{N}, \quad \begin{cases} x_{n+1} = x_n + y_n \\ y_{n+1} = x_n \end{cases} \text{ et avec } x_0 \in \mathbb{N}, y_0 \in \mathbb{N}. \tag{1}$$

1. (INFO) Écrire une fonction en Python `suite(n,x0,y0)` qui prend en argument un entier $n \in \mathbb{N}$ et qui retourne la valeur du couple (x_n, y_n).
2. Montrer que l'on a $\forall n \in \mathbb{N}, \quad x_{n+2} = x_{n+1} + x_n$.
3. Exprimer pour tout $n \in \mathbb{N}$, x_n et y_n en fonction de n, x_0 et y_0.
4. En supposant qu'au temps 0, il y a uniquement une cellule immature, montrer que pour tout $n \in \mathbb{N}$, $x_n = \frac{1}{\sqrt{5}}(r_1^n - r_2^n)$ où $r_1 = \frac{1+\sqrt{5}}{2}$, $r_2 = \frac{1-\sqrt{5}}{2}$.
5. Montrer que l'on a $\forall n \in \mathbb{N}, \quad x_n = \frac{1}{2^{n-1}} \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n}{2k+1} 5^k$.

B - Un modèle général et quelques applications

Dans cette partie, on présente le modèle d'une population structurée en deux classes de manière générique. On appliquera alors ce modèle à différents problèmes issues de la biologie.

Considèrons deux populations $(x_n)_{n\in\mathbb{N}}$ et $(y_n)_{n\in\mathbb{N}}$ évoluant de la manière suivante :

$$\forall n \in \mathbb{N}, \quad \begin{cases} x_{n+1} = ax_n + by_n \\ y_{n+1} = cx_n + dy_n \end{cases} \tag{E}$$

où a, b, c, d sont des paramètres réels.

Par exemple, pour la population de cellules étudiées dans la partie A on a deux classes, l'une correspondant aux cellules matures et l'autre aux cellules immatures. Les paramètres ont dans ce contexte la signification biologique suivante :

— $a = 1$ est la proportion de cellules matures restant matures entre deux temps consécutifs ;
— $b = 1$ est la proportion de cellules immatures devenant matures entre deux temps consécutifs ;
— $c = 1$ est le nombre de cellules immatures obtenues par division d'une cellule mature ;
— $d = 0$ car les cellules immatures ne se divisent pas.

B1 - Étude du modèle général

6. (INFO) Écrire une fonction en Python `suite(n,a,b,c,d,x0,y0)` qui prend en argument un entier $n \in \mathbb{N}$, quatre paramètres réels a, b, c, d et les conditions initiales x_0, y_0 et qui retourne la valeur du couple (x_n, y_n).
7. Montrer que pour tout $n \in \mathbb{N},\ x_{n+2} = (a+d)x_{n+1} + (bc - ad)x_n$.
   Indication : on pourra d'abord montrer que pour tout $n \in \mathbb{N}$, $bdy_n = dx_{n+1} - adx_n$.
8. De manière similaire, montrer que pour tout $n \in \mathbb{N}, y_{n+2} = (a+d)y_{n+1} + (bc - ad)y_n$.
9. Réciproquement, on suppose que les suites $(x_n)_{n\in\mathbb{N}}$ et $(y_n)_{n\in\mathbb{N}}$ vérifient :
   $$\forall n \in \mathbb{N}, x_{n+2} = (a+d)x_{n+1} + (bc-ad)x_n,\ \ y_{n+2} = (a+d)y_{n+1} + (bc-ad)y_n,$$
   et avec comme conditions initiales $x_0, y_0, x_1 = ax_0 + by_0, y_1 = cx_0 + dy_0$.
   Montrer par récurrence que les suites $(x_n)_{n\in\mathbb{N}}$ et $(y_n)_{n\in\mathbb{N}}$ vérifient le système (E). (Indication : on pourra vérifier $P(0)$ et $P(1)$, et supposer que $P(n-1)$ et $P(n)$ sont vraies pour un certain $n \in \mathbb{N}^\star$).
10. On suppose que $(a+d)^2 - 4(ad-bc) = 0$. Pour tout $n \in \mathbb{N}$, exprimer x_n en fonction de n et des autres paramètres.

B2 - Application : une espèce vouée à disparaître

On considère une espèce partagée en deux classes $(x_n)_{n\in\mathbb{N}}$ (les reproducteurs) et $(y_n)_{n\in\mathbb{N}}$ (les parasites). On suppose que l'évolution de l'espèce obéit aux règles suivantes :
— un reproducteur donne naissance à un reproducteur et à deux parasites entre le temps n et $n+1$;
— un reproducteur reste reproducteur entre le temps n et $n+1$;
— un parasite mange deux reproducteurs entre le temps n et $n+1$.
En modélisant à l'aide d'une récurrence, on a alors :

$$\forall n \in \mathbb{N}, \quad \begin{cases} x_{n+1} = 2y_n \\ y_{n+1} = -2x_n + 2y_n \end{cases} \tag{2}$$

On suppose qu'au temps 0, il n'y a pas de parasites et 2 reproducteurs.

11. Exprimer pour tout $n \in \mathbb{N}$, x_n et y_n en fonction de n.
12. Montrer qu'il existe des entiers $n \in \mathbb{N}$ tels que x_n est négatif.

On constate alors qu'à partir d'un certain temps, ce modèle n'est plus valide concernant la population étudiée : en effet, si on obtient $x \leqslant 0$ ceci signifierait qu'il y aurait un nombre négatif d'individus ce qui n'a pas de sens biologique. Dans le cas où on aurait $x_{n_0} \leqslant 0$ pour un certain $n_0 \in \mathbb{N}$, on en conclut que la population $(x_n)_{n\in\mathbb{N}}$ s'éteint à partir d'un certain moment.

B3 - Un modèle tue-mouche

On place dans un même espace confiné deux espèces de mouches, Drosophila simulans et Drosophila melanogaster. On suppose que leur taux de croissance est identique et est de $a > 0$, qu'une D. melanogaster élimine systématiquement $b > 0$ D. simulans entre les instants n et $n+1$. On note respectivement $(x_n)_{n\in\mathbb{N}}$ et $(y_n)_{n\in\mathbb{N}}$ les suites de populations de D. simulans et de D. melanogaster. On suppose que $x_0 > 0$ et $y_0 > 0$.

13. Déterminer les relations de récurrences que vérifient $(x_n)_{n\in\mathbb{N}}$ et $(y_n)_{n\in\mathbb{N}}$.
14. Exprimer x_n et y_n en fonction de x_0, y_0, a, b, n.
15. En déduire que la population de simulans disparaît à partir d'un temps n_0, que l'on exprimera en fonction de a, b, x_0, y_0.
16. (INFO) On suppose que la fonction `tueMouche(x0,y0,a,b,n)` retourne la valeur du couple (x_n, y_n). On considère les deux programmes suivants :

```
n=0
simulans=tueMouche(x0,y0,a,b,n)[0]
while(simulans>0) :
  n=n+1
  simulans=tueMouche(x0,y0,a,b,n)[0]
print(n)
```

```
n=0
melanogaster=tueMouche(x0,y0,a,b,n)[1]
while(simulans>0) :
  n=n+1
  melanogaster=tueMouche(x0,y0,a,b,n)[1]
print(n)
```

   Le ou lesquels de ces programmes terminent ? On se placera dans les conditions de l'énoncé. Dans le cas où il y a une fin au programme, dire ce qui sera affiché.

C - Une population structurée en trois classes

On étudie l'évolution d'une population d'insectes que l'on partage en trois classes d'âges : les larves, les insectes adultes et les insectes âgés. Pour tout $n \in \mathbb{N}$, on note respectivement x_n, y_n et z_n le nombre de larves, adultes et d'insectes âgés au temps n.

On suppose que l'évolution de ces populations obéit à la récurrence

$$\forall n \in \mathbb{N}, \quad \begin{cases} x_{n+1} = 80y_n \\ y_{n+1} = \frac{x_n}{20} \\ z_{n+1} = \frac{y_n}{2} \end{cases}$$

17. Donner une interprétation biologique des coefficients 80, $\frac{1}{2}$, $\frac{1}{20}$, et de la non-apparition de z_n dans cette récurrence.
18. (INFO) Ércire une fonction `popuplation(n)` qui prend en argument une entier $n \in \mathbb{N}$ et qui retourne le nombre total d'insectes.
19. On cherche à établir une relation de récurrence pour la suite $(x_n)_{n\in\mathbb{N}}$. Montrer que pour tout $n \in \mathbb{N}$,
$$x_{n+2} = 4x_n. \tag{3}$$
20. On suppose que pour $n = 0$, il y a 100 larves et pas d'insectes adultes et âgés. Déterminer la population totale en fonction de n.
21. Ce modèle vous paraît-il adapté pour l'étude d'une population d'insectes ?

Remarque : Le modèle présenté, communément appelé modèle de Leslie, se généralise à une structuration en n classes. On obtient alors dans ce cas un système de n récurrences linéaires pouvant être étudié à l'aide du calcul matriciel. Il peut être utilisé dans divers contextes comme en biologie de la conservation ou en démographie humaine.

Problème 2 (Méthodes de calcul et raisonnement)

La ferme expérimentale de Grignon souhaite mettre en place une nouvelle salle de traite automatique donnant plus de liberté aux vaches et sans avoir besoin de chien «super-intelligent» pour diriger le troupeau. Cette nouvelle salle contient n stalles alignées numérotées de 1 à n (où $n \in \mathbb{N}^\star$ est fixé). Dès que les stalles sont libres un sas s'ouvre sur un enclos d'attente pour laisser passer n vaches l'une après l'autre. Chaque vache a une stalle de traite préférée. Pour chaque $k \in [\![1, n]\!] = \{1, 2, 3, \dots, n\}$, on note $s_k \in [\![1, n]\!]$ le numéro de stalle préférée de la k-ième vache arrivant dans la salle de traite. Si la stalle numéro s_k est déjà occupée, la k-ième vache se déplacera jusqu'à la prochaine stalle libre. Les problèmes surviennent lorsqu'aucune stalle restante n'est disponible puisque les vaches ne peuvent pas faire demi-tour.

L'application $k \mapsto s_k$ est simplement représentée par le n-uplet $s = (s_1, s_2, \dots, s_n) \in [\![1, n]\!]^n$ et on dit que s est une fonction de parcage si les n vaches se répartissent dans la salle de traite sans problèmes, c'est-à-dire si chacune occupe une stalle avant le début de la traite. Par exemple pour $n = 6$: $(3, 5, 1, 2, 3, 2)$ est une fonction de parcage au contraire de $(2, 5, 6, 5, 1, 2)$ (voir les figures 1 et 2). De plus, on dit que s est croissante si l'application $k \mapsto s_k$ est croissante, c'est-à-dire si $s_1 \leqslant s_2 \leqslant \cdots \leqslant s_n$.

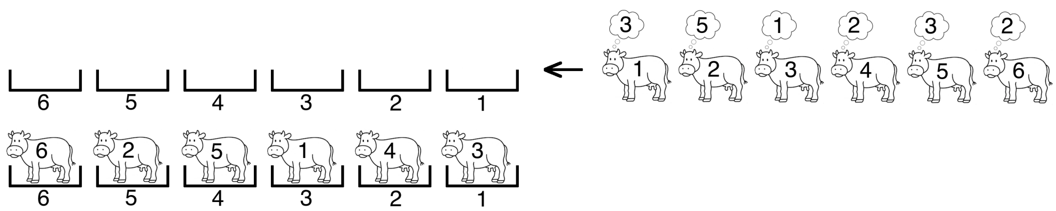

Figure 1 – $(3, 5, 1, 2, 3, 2)$ est une fonction de parcage.

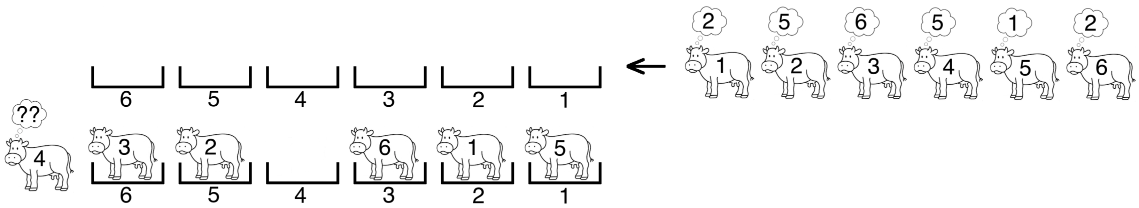

Figure 2 – $(2, 5, 6, 5, 1, 2)$ n'est pas une fonction de parcage.

On note $\mathcal{P}_n$ l'ensemble des fonctions de parcage et $\mathcal{P}_n^+$ le sous-ensemble des fonctions de parcage croissantes. Le but de ce problème est de dénombrer ces deux ensembles.

1. (a) Rappeler le nombre total d'applications de $[\![1, n]\!]$ dans $[\![1, n]\!]$.
   (b) Montrer que $\mathcal{P}_2 = \{(1, 1), (1, 2), (2, 1)\}$.
   (c) Donner, sans justifier, les éléments de $\mathcal{P}_3$ et en déduire que $\mathrm{card}(\mathcal{P}_3) = 16$.
2. On note Θ l'application qui à chaque $s \in [\![1, n]\!]^n$ associe le n-uplet ayant les mêmes composantes que s réordonnées dans l'ordre croissant, par exemple pour $n = 6$: $\Theta(3, 5, 1, 2, 3, 2) = (1, 2, 2, 3, 3, 5)$.
   (a) Discuter brièvement de l'injectivité et de la surjectivité de l'application $\Theta : [\![1, n]\!]^n \to [\![1, n]\!]^n$.
   (b) Dans les deux questions suivantes, on fixe $s = (s_1, s_2, \dots, s_n) \in [\![1, n]\!]^n$ et on note $t = (t_1, t_2, \dots, t_n)$ l'image de s par Θ. Ainsi on a $t_1 \leqslant t_2 \leqslant \cdots \leqslant t_n$.
      i. Pour cette question, on suppose que $t_k > k$ pour un certain $k \in [\![1, n]\!]$. En observant les vaches ayant les préférences $t_k, t_{k+1}, \dots, t_n$, justifier que s n'est pas une fonction de parcage.

ii. Pour cette question, on suppose que s n'est pas une fonction de parcage et on note k le plus petit numéro des stalles qui restent inoccupées. En observant les vaches ayant les préférences $t_1, t_2, \ldots, t_k$, justifier par l'absurde que $t_k > k$.

(c) En déduire la description suivante de l'ensemble des fonctions de parcage :

$$\boxed{\mathcal{P}_n = \left\{ s \in [\![1,n]\!]^n \mid \forall k \in [\![1,n]\!],\ t_k \leqslant k \text{ où } (t_1, t_2, \ldots, t_n) = \Theta(s) \right\}}.$$

3. Pour chaque $t = (t_1, t_2, \ldots, t_n) \in [\![1,n]\!]^n$ croissant, c'est-à-dire tel que $t_1 \leqslant t_2 \leqslant \cdots \leqslant t_n$, on note $\mathcal{A}_t$ l'ensemble des antécédents de t par l'application Θ, c'est-à-dire : $\mathcal{A}_t = \{s \in [\![1,n]\!]^n \mid \Theta(s) = t\}$. Par définition de l'application Θ, $\mathcal{A}_t$ est donc l'ensemble des n-uplets ayant les mêmes composantes que t dans un ordre quelconque, autrement dit $\mathcal{A}_t$ est l'ensemble des anagrammes de t.

(a) Montrer que si $t \in [\![1,n]\!]^n$ et $t' \in [\![1,n]\!]^n$ sont distincts et croissants, alors $\mathcal{A}_t \cap \mathcal{A}_{t'} = \emptyset$.

(b) À l'aide du résultat de la question 2, montrer que si $t \in \mathcal{P}_n^+$, alors $\mathcal{A}_t \subset \mathcal{P}_n$.

(c) Prouver que l'union des ensembles $\mathcal{A}_t$ où t décrit $\mathcal{P}_n^+$ est égal à $\mathcal{P}_n$ et en déduire que

$$\boxed{\operatorname{card}(\mathcal{P}_n) = \sum_{t \in \mathcal{P}_n^+} \operatorname{card}(\mathcal{A}_t)}.$$

4. (a) Écrire $\mathcal{P}_2$ sous la forme $\bigcup_{t \in \mathcal{P}_2^+} \mathcal{A}_t$ en déterminant $\mathcal{P}_2^+$ et chaque ensemble $\mathcal{A}_t$ où t décrit $\mathcal{P}_2^+$.

(b) Même question pour $\mathcal{P}_3$.

(c) Dans les trois questions suivantes, on s'intéresse au cas $n = 4$.

i. À l'aide du résultat de la question 2, donner, sans justifier, les éléments de $\mathcal{P}_4^+$.

ii. Pour chaque $t \in \mathcal{P}_4^+$, déterminer $\operatorname{card}(\mathcal{A}_t)$, c'est-à-dire le nombre d'anagrammes de t.

iii. En déduire que $\operatorname{card}(\mathcal{P}_4) = 125$.

5. Pour chaque $n \in \mathbb{N}^\star$ et chaque $(i,j) \in [\![1,n+1]\!]^2$, on définit les ensembles suivants :

$$\mathcal{E}_n^i = \left\{ (t_1, t_2, \ldots, t_n) \in \mathcal{P}_n^+ \mid t_n = i \right\} \text{ et } \mathcal{F}_{n+1}^{i,j} = \left\{ (t_1, t_2, \ldots, t_n, t_{n+1}) \in \mathcal{P}_{n+1}^+ \mid (t_n, t_{n+1}) = (i,j) \right\}.$$

De plus, on note E_n^i le cardinal de $\mathcal{E}_n^i$.

(a) Calculer E_n^1 et E_n^{n+1}.

(b) Dans les quatre questions suivantes, on fixe $(i,j) \in [\![1,n+1]\!]^2$ tel que $i \leqslant j$ et on pose :

$$\varphi : \mathcal{E}_n^i \to \mathcal{F}_{n+1}^{i,j},\ (t_1, t_2, \ldots, t_n) \mapsto (t_1, t_2, \ldots, t_n, j).$$

i. Justifier que si $(t_1, t_2, \ldots, t_n) \in \mathcal{P}_n^+$, alors $(t_1, t_2, \ldots, t_n, \ell) \in \mathcal{P}_{n+1}^+$ pour tout $\ell \in [\![t_n, n+1]\!]$.

ii. En déduire que φ est bien définie, c'est-à-dire que $\varphi(t) \in \mathcal{F}_{n+1}^{i,j}$ pour tout $t \in \mathcal{E}_n^i$.

iii. Justifier que si $(t_1, t_2, \ldots, t_n, t_{n+1}) \in \mathcal{P}_{n+1}^+$, alors $(t_1, t_2, \ldots, t_n) \in \mathcal{P}_n^+$.

iv. Montrer que $\varphi : \mathcal{E}_n^i \to \mathcal{F}_{n+1}^{i,j}$ est bijective et que $\operatorname{card}\left(\mathcal{F}_{n+1}^{i,j}\right) = E_n^i$.

(c) Pour cette question, on fixe $j \in [\![1,n+1]\!]$. Prouver que $\bigcup_{i=1}^{j} \mathcal{F}_{n+1}^{i,j} = \mathcal{E}_{n+1}^j$ et en déduire que

$$E_{n+1}^j = \sum_{i=1}^{j} E_n^i.$$

(d) Montrer que

$$\forall (p,q) \in \mathbb{N}^2,\ \sum_{\ell=0}^{p} \binom{q+\ell}{\ell} = \binom{p+q+1}{p}$$

(indication : utiliser après l'avoir justifié que $\binom{q+\ell}{\ell} = \binom{q+\ell+1}{\ell} - \binom{q+\ell}{\ell-1}$ pour tout $\ell \in \mathbb{Z}$).

(e) À l'aide des résultats précédents, démontrer par récurrence que pour tout $n \in \mathbb{N}^\star$ on a :

$$\forall i \in [\![1,n]\!],\ E_n^i = \binom{n+i-2}{i-1} - \binom{n+i-2}{i-2}.$$

(f) Justifier brièvement que $\mathcal{P}_n^+$ est égal à l'union disjointe des $\mathcal{E}_n^1, \mathcal{E}_n^2, \ldots, \mathcal{E}_n^n$, puis en déduire que

$$\operatorname{card}\left(\mathcal{P}_n^+\right) = \sum_{i=1}^{n} E_n^i = \binom{2n-1}{n-1} - \binom{2n-1}{n-2}.$$

(g) Conclure que

$$\boxed{\operatorname{card}\left(\mathcal{P}_n^+\right) = \frac{1}{n+1}\binom{2n}{n}}.$$

6. (a) À l'aide du résultat précédent, déterminer le nombre d'éléments de $\mathcal{P}_5^+$.

(b) En s'inspirant de la question 4.(c), expliquer comment on pourrait déterminer $\operatorname{card}(\mathcal{P}_5)$.