Olympiades académiques

Corse 2024

Exercice 1 : Les ploutons

Partie A : Les petits ploutons

L'algorithme « plouton » dresse, par exhaustivité, la liste des nombres ploutons qui sont inférieurs ou égaux à un nombre donné.
 	
2. Un nombre est premier si et seulement s'il a exactement deux diviseurs. Seul le plus petit d'entre eux est un plouton puisque les nombres premiers suivants n'ont pas plus de diviseurs que lui.
	<pre>>>> def plouton (n):
	        print("1 est le premier plouton, il a un seul diviseur")
	        d=1
	        k=1
	        for x in range(2,n+1):
	                u=0
	                for y in range(1,x+1):
	                        if x%y==0:
	                                u=u+1
	                if u>d:
	                        d=u
	                        k=k+1
	                        print("Le plouton numéro",k,"est",x,"qui a",d,"diviseurs")
	
	>>> plouton(200)
	1 est le premier plouton, il a un seul diviseur
	Le plouton numéro 2 est 2 qui a 2 diviseurs
	Le plouton numéro 3 est 4 qui a 3 diviseurs
	Le plouton numéro 4 est 6 qui a 4 diviseurs
	Le plouton numéro 5 est 12 qui a 6 diviseurs
	Le plouton numéro 6 est 24 qui a 8 diviseurs
	Le plouton numéro 7 est 36 qui a 9 diviseurs
	Le plouton numéro 8 est 48 qui a 10 diviseurs
	Le plouton numéro 9 est 60 qui a 12 diviseurs
	Le plouton numéro 10 est 120 qui a 16 diviseurs
	Le plouton numéro 11 est 180 qui a 18 diviseurs</pre>

Il faut attendre le plouton numéro 15 pour obtenir un nombre qui a un autre facteur premier que 2, 3 ou 5.	<pre>>>> plouton(1000)
	1 est le premier plouton, il a un seul diviseur
	Le plouton numéro 2 est 2 qui a 2 diviseurs
	Le plouton numéro 3 est 4 qui a 3 diviseurs
	Le plouton numéro 4 est 6 qui a 4 diviseurs
	Le plouton numéro 5 est 12 qui a 6 diviseurs
	Le plouton numéro 6 est 24 qui a 8 diviseurs
	Le plouton numéro 7 est 36 qui a 9 diviseurs
	Le plouton numéro 8 est 48 qui a 10 diviseurs
	Le plouton numéro 9 est 60 qui a 12 diviseurs
	Le plouton numéro 10 est 120 qui a 16 diviseurs
	Le plouton numéro 11 est 180 qui a 18 diviseurs
	Le plouton numéro 12 est 240 qui a 20 diviseurs
	Le plouton numéro 13 est 360 qui a 24 diviseurs
	Le plouton numéro 14 est 720 qui a 30 diviseurs
	Le plouton numéro 15 est 840 qui a 32 diviseurs</pre>

Partie B : Les ploutons inférieurs ou égaux à 200

1. Outre 1 et 2, nous avons successivement : $4 = 2^2$; $6 = 2 \times 3$; $12 = 2^2 \times 3$; $24 = 2^3 \times 3$; $36 = 2^2 \times 3^2$.

2. C'est faux, par exemple $8 = 2^3$; $30 = 2 \times 3 \times 5$ ne sont pas plutons. Le nombre 8 a 4 diviseurs, autant que 6, et 30 a 8 diviseurs, autant que 24.

3. Laissée au lecteur

4. C'est faux, comme le montre le contre-exemple 7.

5. Rappelons le théorème suivant : Soit N un entier strictement positif. Un entier d divise N si et seulement si chaque facteur premier de d est un facteur premier de N affecté d'un exposant inférieur ou égal à celui dont il est affecté dans la décomposition de N.

Ainsi, que $2^a \times 3^b \times 5^c$ soit plouton ou non, un nombre d divise $2^a \times 3^b \times 5^c$ si et seulement si sa décomposition en produit de facteurs premiers est de la forme $d = 2^x \times 3^y \times 5^z$ avec x, y, z entiers tels que $0 \leq x \leq a$; $0 \leq y \leq b$; $0 \leq z \leq c$.

L'application : $(x\,;y\,;z) \in \{0\,;\ldots;a\} \times \{0\,;\ldots;b\} \times \{0\,;\ldots;c\} \longmapsto 2^x \times 3^y \times 5^z$ est une application surjective de $\{0\,;\ldots;a\} \times \{0\,;\ldots;b\} \times \{0\,;\ldots;c\}$ sur l'ensemble des diviseurs de $2^a \times 3^b \times 5^c$. Mais d'après l'unicité de la décomposition d'un nombre en facteurs premiers, cette application est aussi injective (deux triplets distincts ont pour images des diviseurs distincts). L'ensemble des diviseurs de $2^a \times 3^b \times 5^c$ a le même nombre d'éléments que l'ensemble $\{0\,;\ldots;a\} \times \{0\,;\ldots;b\} \times \{0\,;\ldots;c\}$. Il a donc $(a+1) \times (b+1) \times (c+1)$ éléments.

6. Pour a, b, c entiers naturels, comparons $2^a \times 3^b$ avec $2^b \times 3^a$ et $3^b \times 5^c$ avec $3^c \times 5^b$.

D'une part $\frac{2^b \times 3^a}{2^a \times 3^b} = \left(\frac{2}{3}\right)^{b-a}$ et d'autre part $\frac{3^c \times 5^b}{3^b \times 5^c} = \left(\frac{3}{5}\right)^{c-b}$.

Or, les nombres $\frac{2}{3}$ et $\frac{3}{5}$ sont strictement positifs et plus petits que 1. Il en est de même de toute puissance strictement positive de ces nombres.

Supposons que $b > a$. Alors $\frac{2^b \times 3^a}{2^a \times 3^b} < 1$. Donc $2^b \times 3^a < 2^a \times 3^b$. Les deux entiers $2^b \times 3^a$ et $2^a \times 3^b$ ont le même nombre de diviseurs, soit $(a+1) \times (b+1)$, et le plus petit des deux nombres est strictement $2^b \times 3^a$.

Supposons que $c > b$. Alors $\frac{3^c \times 5^b}{3^b \times 5^c} < 1$. Donc $3^c \times 5^b < 3^b \times 5^c$. Les deux entiers $3^c \times 5^b$ et $3^b \times 5^c$ ont le même nombre de diviseurs, soit $(b+1) \times (c+1)$, et le plus petit des deux nombres est strictement $3^c \times 5^b$.

Supposons maintenant que le nombre $2^a \times 3^b \times 5^c$ soit un plouton.

- Nécessairement, $a \geq b$ car dans le cas contraire $2^b \times 3^a \times 5^c$ aurait le même nombre de diviseurs et serait strictement plus petit.
- Nécessairement, $b \geq c$ car dans le cas contraire $2^a \times 3^c \times 5^b$ aurait le même nombre de diviseurs et serait strictement plus petit.

En définitive, si $2^a \times 3^b \times 5^c$ est un plouton, alors $a \geq b \geq c$.

7. Le nombre $30 = 2 \times 3 \times 5$ admet 8 diviseurs mais n'est pas plouton car 24 a aussi 8 diviseurs et est plus petit que lui. La réciproque de l'implication de la question 6 est fausse.

8. Nous laissons le lecteur traiter cette question.

Les plus grands exposants a, b, c possibles sont $a = 7\,;b = 4\,;c = 3$.

L'algorithme « ploutonpot » affiche ci-contre les ploutons potentiels. L'algorithme en affiche 20 et non 21 comme l'énoncé l'indique. Je ne sais pas lequel a échappé, le lecteur le découvrira probablement …

9. Traitée par l'algorithme « plouton ».

```
>>> def ploutonpot():
        for c in range(0,3):
            for b in range(c,4):
                for a in range(b,7):
                    n=(2**a)*(3**b)*(5**c)
                    if n<=200:
                        print(n)

>>> ploutonpot()
1
2
4
8
16
32
64
6
12
24
48
96
192
36
72
144
30
60
120
180
```

Exercice 2 : Parcours gagnants

Partie I

1. Il y a $3 \times 3 \times 3 = 27$ parcours possibles.

2. L'algorithme « parcours » écrit sur le logiciel TI-Nspire (qui nous paraît en cette circonstance plus ergonomique que Python) décrit en extension les 27 parcours possibles. Sont affichés ci-contre les 9 parcours qui commencent par le vecteur $\vec{u}$. On relève que 5 de ces parcours sont gagnants.

$\begin{bmatrix}1&1\\2&2\\3&3\end{bmatrix}$ perdant

$\begin{bmatrix}1&1\\2&2\\3&2\end{bmatrix}$ perdant

$\begin{bmatrix}1&1\\2&2\\3&1\end{bmatrix}$ perdant

$\begin{bmatrix}1&1\\2&1\\3&2\end{bmatrix}$ perdant

$\begin{bmatrix}1&1\\2&1\\3&1\end{bmatrix}$ gagnant

$\begin{bmatrix}1&1\\2&1\\3&0\end{bmatrix}$ gagnant

$\begin{bmatrix}1&1\\2&0\\3&1\end{bmatrix}$ gagnant

$\begin{bmatrix}1&1\\2&0\\3&0\end{bmatrix}$ gagnant

$\begin{bmatrix}1&1\\2&0\\3&-1\end{bmatrix}$ gagnant

```
parcours
Define parcours()=
Prgm
Define pas=[[1,1],[1,0],[1,-1]]
Define marche=[[0,0],[0,0],[0,0]]
Define x=0
For i,1,3
 pas[i]→marche[1]
 For j,1,3
  marche[1]+pas[j]→marche[2]
  For k,1,3
   marche[2]+pas[k]→marche[3]
      If max(|marche[2,2]|,|marche[3,2]|)≤1 Then
       x+1→x
       Disp marche,"gagnant"
      Else
        Disp marche,"perdant"
      EndIf
  EndFor
 EndFor
EndFor
Disp "nombre de gagnants",x
EndPrgm
```

3. Ici les 9 parcours qui commencent par le vecteur $\vec{v}$. On relève que 7 de ces parcours sont gagnants.

4. Puis les 9 parcours qui commencent par le vecteur $\vec{w}$. Symétriquement à ceux qui commencent par $\vec{u}$, 5 de ces parcours sont gagnants. Parmi les 27 parcours possibles, au total 17 sont gagnants.

Nous sommes en situation d'équiprobabilité.

La probabilité qu'un parcours choisi au hasard soit gagnant est $\frac{17}{27}$.

$\begin{bmatrix}1&0\\2&1\\3&2\end{bmatrix}$ perdant

$\begin{bmatrix}1&0\\2&1\\3&1\end{bmatrix}$ gagnant

$\begin{bmatrix}1&0\\2&1\\3&0\end{bmatrix}$ gagnant

$\begin{bmatrix}1&0\\2&0\\3&1\end{bmatrix}$ gagnant

$\begin{bmatrix}1&0\\2&0\\3&0\end{bmatrix}$ gagnant

$\begin{bmatrix}1&0\\2&0\\3&-1\end{bmatrix}$ gagnant

$\begin{bmatrix}1&0\\2&-1\\3&0\end{bmatrix}$ gagnant

$\begin{bmatrix}1&0\\2&-1\\3&-1\end{bmatrix}$ gagnant

$\begin{bmatrix}1&0\\2&-1\\3&-2\end{bmatrix}$ perdant

$\begin{bmatrix}1&-1\\2&0\\3&1\end{bmatrix}$ gagnant

$\begin{bmatrix}1&-1\\2&0\\3&0\end{bmatrix}$ gagnant

$\begin{bmatrix}1&-1\\2&0\\3&-1\end{bmatrix}$ gagnant

$\begin{bmatrix}1&-1\\2&-1\\3&0\end{bmatrix}$ gagnant

$\begin{bmatrix}1&-1\\2&-1\\3&-1\end{bmatrix}$ gagnant

$\begin{bmatrix}1&-1\\2&-1\\3&-2\end{bmatrix}$ perdant

$\begin{bmatrix}1&-1\\2&-2\\3&-1\end{bmatrix}$ perdant

$\begin{bmatrix}1&-1\\2&-2\\3&-2\end{bmatrix}$ perdant

$\begin{bmatrix}1&-1\\2&-2\\3&-3\end{bmatrix}$ perdant

Partie B : De proche en proche

1. $a_1 = b_1 = c_1$ et $T_1 = 3$.

2. Remarque préliminaire. La symétrie axiale d'axe *Ox* échange les pas de vecteur $\vec{u}$ avec ceux de vecteur par $\vec{w}$. La situation étudiée est ainsi globalement invariante par cette symétrie axiale. Cette invariance justifie le fait qu'il y a autant de parcours qui se terminent en un point d'ordonnée 1 que de parcours qui se terminent en un point d'ordonnée -1. Autrement dit, $a_n = c_n$.

2.a et b. Chacun des a_n parcours atteignant le point $(n\,;1)$ se prolonge d'une seule façon vers le point $(n+1\,;1)$ par un pas de vecteur $\vec{v}$ et se prolonge d'une seule façon vers le point $(n+1\,;0)$ par un pas de vecteur $\vec{w}$.

Chacun des b_n parcours atteignant le point $(n\,;0)$ se prolonge d'une seule façon vers le point $(n+1\,;1)$ par un pas de vecteur $\vec{u}$, se prolonge d'une seule façon vers le point $(n+1\,;0)$ par un pas de vecteur $\vec{v}$ et se prolonge d'une seule façon vers le point $(n+1\,;-1)$ par un pas de vecteur $\vec{w}$

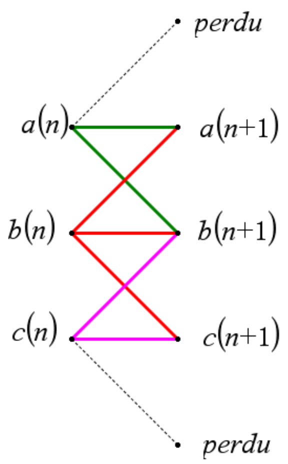
Chacun des c_n parcours atteignant le point $(n\,;-1)$ se prolonge d'une seule façon vers le point $(n+1\,;-1)$ par un pas de vecteur $\vec{v}$ et se prolonge d'une seule façon vers le point $(n+1\,;0)$ par un pas de vecteur $\vec{u}$.

Les mouvements entre deux états consécutifs peuvent être schématisés comme ci-contre. Nous obtenons les relations :

$$\begin{cases} a_{n+1} = a_n + b_n \\ b_{n+1} = a_n + b_n + c_n \\ c_{n+1} = b_n + c_n \end{cases}$$

Compte tenu de la relation admise $a_n = c_n$ pour tout entier $n > 0$ (que l'on pourrait cependant démontrer par récurrence) :

$$\begin{cases} a_{n+1} = a_n + b_n \\ b_{n+1} = 2a_n + b_n \end{cases}$$



En ce qui concerne les sommes des trois nombres, compte tenu de la relation admise :

$T_n = a_n + b_n + c_n = a_n + 2b_n$ pour tout entier $n > 0$.

Soit alors un entier $n > 0$.

D'une part : $T_{n+2} = a_{n+2} + 2b_{n+2} = (a_{n+1} + b_{n+1}) + 2(2a_{n+1} + b_{n+1}) = 5a_{n+1} + 3b_{n+1}$

Soit : $T_{n+2} = 5(a_n + b_n) + 3(2a_n + b_n) = 11a_n + 8b_n$

D'autre part : $2T_{n+1} + T_n = 2(a_{n+1} + 2b_{n+1}) + (a_n + 2b_n) = 2(a_n + b_n) + 4(2a_n + b_n) + (a_n + 2b_n)$

Soit : $2T_{n+1} + T_n = 11a_n + 8b_n$

Nous sommes parvenus à deux expressions identiques, par conséquent : $T_{n+2} = 2T_{n+1} + T_n$

Partie C : Expression explicite

1. Nous déléguons la question au logiciel de calcul formel TI-Nspire, voir ci-contre.	$\text{solve}(x^2=2\cdot x+1,x)$	$x=-(\sqrt{2}-1)$ or $x=\sqrt{2}+1$
	Define $x1=-(\sqrt{2}-1)$	*Terminé*
	Define $x2=\sqrt{2}+1$	*Terminé*
2.a. Nous avons vérifié que $u_1 = 3 = T_1$	Define $u(n)=\frac{(1-\sqrt{2})\cdot x1^n}{2}+\frac{(1+\sqrt{2})\cdot x2^n}{2}$	*Terminé*
	$u(1)$	3
	$u(2)$	7

Cependant, cette vérification ne suffit pas pour initialiser la démonstration par récurrence qui va suivre, car la relation de récurrence liant les termes de la suite (T_n) porte non pas sur un rang mais sur deux rangs.

Nous vérifions donc aussi d'une part que $u_2 = 7$ et que d'autre part le nombre T_2 de parcours de deux pas gagnants est bien lui aussi égal à 7 (sur 9 parcours possibles). Le lecteur le vérifiera sans difficulté.

2.b. Démontrons par récurrence que la propriété $\mathcal{P}_n$: « $T_n = u_n$ » est vérifiée pour tout entier strictement positif *n*.

Initialisation : Nous avons vérifié que $\begin{cases} T_1 = u_1 = 3 \\ T_2 = u_2 = 7 \end{cases}$

Hérédité : Supposons que pour un certain entier strictement positif *n* : $\begin{cases} T_n = u_n \\ T_{n+1} = u_{n+1} \end{cases}$.

La relation de récurrence porte sur deux rangs, nous devons émettre l'hypothèse de travail sur deux rangs.

Au rang suivant : $T_{n+2} = 2T_{n+1} + T_n = 2u_{n+1} + u_n = 2\left(\frac{1-\sqrt{2}}{2}{x_1}^{n+1} + \frac{1+\sqrt{2}}{2}{x_2}^{n+1}\right) + \left(\frac{1-\sqrt{2}}{2}{x_1}^{n} + \frac{1+\sqrt{2}}{2}{x_2}^{n}\right)$.

Soit : $T_{n+2} = \frac{1-\sqrt{2}}{2}{x_1}^n(2x_1 + 1) + \frac{1+\sqrt{2}}{2}{x_2}^n(2x_2 + 1)$

Vu les propriétés algébriques des deux nombres x_1 et x_2, nous obtenons :

$$T_{n+2} = \frac{1-\sqrt{2}}{2}{x_1}^n({x_1}^2) + \frac{1+\sqrt{2}}{2}{x_2}^n({x_2}^2) = \frac{1-\sqrt{2}}{2}{x_1}^{n+2} + \frac{1+\sqrt{2}}{2}{x_2}^{n+2} = u_{n+2}$$

Ainsi, nous pouvons écrire : $\begin{cases} T_n = u_n \\ T_{n+1} = u_{n+1} \end{cases} \Longrightarrow \begin{cases} T_{n+1} = u_{n+1} \\ T_{n+2} = u_{n+2} \end{cases}$

Conclusion : La propriété « $T_n = u_n$ » est initialisée aux rangs 1 et 2 et elle est héréditaire, elle est vérifiée pour tout entier strictement positif.

L'expression explicite est : $T_n = \frac{1-\sqrt{2}}{2}{x_1}^n + \frac{1+\sqrt{2}}{2}{x_2}^n$

On en déduit que la probabilité qu'un parcours de taille *n* soit gagnant est : $p_n = \dfrac{\frac{1-\sqrt{2}}{2}{x_1}^n + \frac{1+\sqrt{2}}{2}{x_2}^n}{3^n}$ Ci-contre les dix premières valeurs.	$\mathrm{seq}(u(n),n,1,10)$ $\{3,7,17,41,99,239,577,1393,3363,8119\}$ $\mathrm{seq}\left(\frac{u(n)}{3^n},n,1,10\right)$ $\left\{1,\frac{7}{9},\frac{17}{27},\frac{41}{81},\frac{11}{27},\frac{239}{729},\frac{577}{2187},\frac{1393}{6561},\frac{1121}{6561},\frac{8119}{59049}\right\}$ $\mathrm{seq}\left(\frac{u(n)}{3^n},n,1,10\right)$ $\{1.,0.778,0.63,0.506,0.407,0.328,0.264,0.212,0.171,0.137\}$

Remarque. *Cette probabilité s'écrit ainsi :* $p_n = \frac{1-\sqrt{2}}{2}\left(\frac{x_1}{3}\right)^n + \frac{1+\sqrt{2}}{2}\left(\frac{x_2}{3}\right)^n$

Les deux réels $\left(\frac{x_1}{3}\right)$ *et* $\left(\frac{x_2}{3}\right)$ *sont compris strictement entre* -1 *et* 1.

Ainsi, la suite de leurs puissances sont des suites qui tendent vers zéro à l'infini.

Nous pourrions conclure que : $\lim_{n\to\infty} p_n = 0$

La probabilité d'obtenir un parcours indéfiniment gagnant est nulle. Un parcours du type considéré va inexorablement à sa perte, tôt ou tard.