CORRIGÉ DEVOIR MAISON N ° 2 TERMINALE spé

Exercice 1 :

Partie A : On considère la suite numérique (u_n) définie sur l'ensemble des entiers naturels $\mathbb{N}$ par $u_0 = 0$ et pour tout entier naturel n, $u_{n+1} = \dfrac{1}{2-u_n}$.

```
n = 10
u = 0
for k in range(n) :
    u = 1/(2 – u)
    print(u)
```

1. a) Le programme python complété ci-contre pour obtenir en sortie toutes les valeurs de u_n pour n variant de 1 à 10 :

b) Conjecture : la suite numérique (u_n) est croissante et sa limite semble être 1.

2. Démontrons par récurrence que la suite (u_n) est bornée par 0 et 1 : La propriété (P_n) est : $0 \leq u_n \leq 1$.

Initialisation : $u_0 = 0$ et $0 \leq 0 \leq 1$, donc (P_0) est vraie.

Hérédité : On suppose que (P_n) est vraie pour un entier naturel n ; montrons que (P_{n+1}) est vraie :

$0 \leq u_n \leq 1 \Rightarrow -1 \leq -u_n \leq 0$ (la multiplication par – 1 inverse l'ordre) $\Rightarrow 1 \leq 2 - u_n \leq 2 \Rightarrow$

$\dfrac{1}{2} \leq \dfrac{1}{2-u_n} \leq 1$ (le passage à l'inverse de deux nombres positifs inverse l'ordre) $\Rightarrow 0 < \dfrac{1}{2} \leq u_{n+1} \leq 1$.

Donc (P_{n+1}) est vraie.

Conclusion : La propriété (P_n) est initialisée et héréditaire, donc elle est vraie pour tout entier naturel n.

Partie B : On considère la suite numérique (v_n) définie sur l'ensemble des entiers naturels $\mathbb{N}$ par $v_n = \dfrac{1}{1-u_n}$.

1. Pour tout entier naturel n,

$$v_{n+1} = \frac{1}{1-u_{n+1}} = \frac{1}{1-\dfrac{1}{2-u_n}} = \frac{1}{\dfrac{2-u_n-1}{2-u_n}} = \frac{1}{\dfrac{1-u_n}{2-u_n}} = \frac{2-u_n}{1-u_n} = \frac{1-u_n+1}{1-u_n} = 1 + \frac{1}{1-u_n} = 1 + v_n .$$

Donc la suite (v_n) est arithmétique de raison 1 et de premier terme $v_0 = \dfrac{1}{1-u_0} = 1$.

2. Ainsi pour tout entier naturel n, $v_n = 1 + n$.

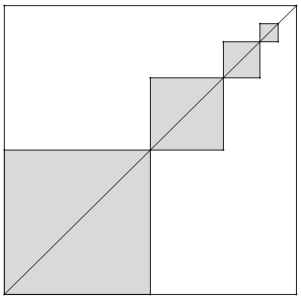
Et $1 + n = \dfrac{1}{1-u_n}$, soit $(1 + n)(1 - u_n) = 1$, d'où $1 - u_n = \dfrac{1}{1+n}$ et $u_n = 1 - \dfrac{1}{1+n} = \dfrac{n}{1+n}$.

3. Variations de (u_n) : $u_{n+1} - u_n = \dfrac{n+1}{2+n} - \dfrac{n}{1+n} = \dfrac{(n+1)^2 - n(n+2)}{(2+n)(1+n)} = \dfrac{n^2+2n+1-n^2-2n}{(2+n)(1+n)} = \dfrac{1}{(2+n)(1+n)}$ qui est strictement positif puisque n est un entier naturel. Donc pour tout entier naturel n, $u_{n+1} - u_n > 0$, donc $u_{n+1} > u_n$ et la suite est strictement croissante.

Limite de (u_n) : $\dfrac{n}{1+n} = \dfrac{n}{n(1+\frac{1}{n})} = \dfrac{1}{1+\frac{1}{n}}$; comme $\lim\limits_{n \to +\infty} \dfrac{1}{n} = 0$, alors $\lim\limits_{n \to +\infty} u_n = 1$.

Exercice 2 :

On considère un carré de côté 1. On construit un carré dont un sommet est le centre du carré comme sur la figure ci-contre. Puis on construit un carré dont le côté est la moitié du précédent, et ainsi de suite tel que un sommet du carré construit est toujours sur la diagonale du premier carré (ces carrés sont grisés sur la figure ci-contre).



On note a_n = l'aire du n-ième carré construit à l'intérieur du carré de côté 1.

Ainsi $a_1 = \dfrac{1}{4}$.

1. Le côté du n-ième carré construit est la moitié du carré précédent, donc son aire est le quart de l'aire du carré précédent, donc $a_{n+1} = \dfrac{1}{4} a_n$; donc la suite (a_n) est géométrique de raison $\dfrac{1}{4}$ et de premier terme $a_1 = \dfrac{1}{4}$;

donc $a_n = \dfrac{1}{4} \times \left(\dfrac{1}{4}\right)^{n-1} = \left(\dfrac{1}{4}\right)^n$.

2. On note S_n la somme des aires des carrés, soit $S_n = \sum_{k=1}^{k=n} a_k$.

$$S_n = \sum_{k=1}^{k=n} a_k = \frac{1}{4} \times \frac{1-\left(\frac{1}{4}\right)^n}{1-\frac{1}{4}} = \frac{1}{4} \times \frac{1-\left(\frac{1}{4}\right)^n}{\frac{3}{4}} = \frac{1}{4} \times \frac{4}{3} \times \left(1-\left(\frac{1}{4}\right)^n\right) = \frac{1}{3} \times \left(1-\left(\frac{1}{4}\right)^n\right).$$

Comme $-1 < \frac{1}{4} < 1$, alors $\lim\limits_{n \to +\infty} \left(\frac{1}{4}\right)^n = 0$, donc $\lim\limits_{n \to +\infty} S_n = \frac{1}{3}$.

3. La plus petite valeur de n telle que $S_n \geq 0{,}333333$ est $n = 10$.

4. L'algorithme (ou le programme python) permettant de trouver la valeur précédente :

Les premiers termes de la suite à 10^{-12} près :

```
u0=0
u1=0.25
u2=0.3125
u3=0.328125
u4=0.33203125
u5=0.3330078125
u6=0.333251953125
u7=0.333312988281
u8=0.33332824707
u9=0.333332061768
u10=0.333333015442
u11=0.33333325386
u12=0.333333313465
u13=0.333333328366
u14=0.333333332092
u15=0.333333333023
```

```
A = 0.333333
u = 1/4
S = u
n = 0
while (S < A) :
    n = n + 1
    u = u/4
    S = S + u
print("le plus petit entier n tel que Sn >= A :", n)
```