Code de partage avec Capytale : a6e6-7631015

On utilisera les bibliothèques suivantes :

```
import numpy as np
import numpy.random as rd
import numpy.linalg as al
import matplotlib.pyplot as plt
```

1 Fonctions, suites récurrentes

1. Définir la fonction `f` associée à $f : x \mapsto (x+1)e^{-x/2}$
2. Définir la fonction `df` associée à la fonction dérivée de f. On en donnera une forme factorisée.
3. Définir la fonction `d2f` associée à la dérivée seconde de f. On en donnera une forme factorisée.
4. Tracer la courbe représentative de f' sur $[0,5]$
5. Justifier que $|f'(x)| \leqslant \max(f'(0), |f'(3)|))$ pour $x \geqslant 0$
6. A l'aide d'une dichotomie, donner un encadrement à 10^{-2} près de l'unique solution ℓ de l'équation $f(x) = x$ sur $\mathbb{R}_+$
   On commencera par déterminer l'intervalle d'étude.
7. Définir la suite récurrente : $u_0 = 0$ et, pour $n \in \mathbb{N}$: $u_{n+1} = f(u_n)$
8. On admet que : pour tout $n \in \mathbb{N}$, $|u_n - \ell| \leqslant 2^{-n+1}$
   A l'aide de cette majoration, déterminer un entier n_0 pour lequel u_n est une valeur approchée de ℓ à 10^{-5} près pour tout entier $n \geqslant n_0$

2 Suites récurrentes d'ordre 2

On considère la suite $(u_n)_{n\in\mathbb{N}}$ définie par $u_0 = 0$, $u_1 = \dfrac{1}{16}$ et $\forall n \in \mathbb{N}, \quad u_{n+2} = \frac{3}{4}u_{n+1} + \frac{3}{16}u_n$

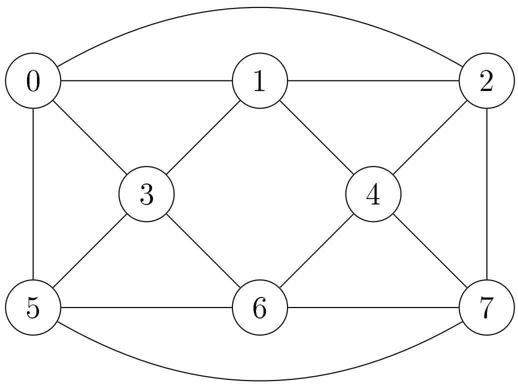
1. Définir une fonction `PremiersTermes(n)` qui renvoie la listes des $n+1$ premiers termes de la suite.
   Exemple : `PremiersTermes(4)` doit renvoyer :
   `[0, 0.0625, 0.046875, 0.046875, 0.0439453125]`
2. Vérifier numériquement que : $\displaystyle\sum_{n=0}^{+\infty} u_n = 1$
3. On admet qu'il existe une variable aléatoire X admettant un moment d'ordre 2, à valeurs dans $\mathbb{N}$, telle que $\forall n \in \mathbb{N}, \quad P(X = n) = u_n$
   Calculer des valeurs approchées de $E(X)$ et $V(X)$ en calculant des sommes partielles.
4. Compléter la fonction Python suivante permettant de calculer un terme de la suite u par une fonction récursive :

```
def SR(u0, u1, n):
    if ... :
        return ...
    elif ...:
        return ...
    else:
```

```
        return ...
    print(SR(...........)) # pour tester
```

3 Graphes

On considère le graphe **G** suivant :



1. Définir la liste d'adjacence `L` de ce graphe.
2. Définir la matrice d'adjacence `A` de ce graphe.
3. Déterminer le nombre de chaînes de longueur 4 allant du sommet 0 au sommet 2

4 Probabilités

Une urne contient n boules noires et b boules blanches, avec $n \geqslant 1$ et $b \geqslant 1$
On effectue une succession de tirages dans l'urne avec remise.
On note X_k le rang d'apparition de la $k^{\text{ème}}$ boule blanche pour $k \in \mathbb{N}^*$
Ainsi, X_1 est le rang d'apparition de la première boule blanche, X_2 celui de la deuxième boule blanche, etc.
On pose, pour $i \in \mathbb{N}^*$, T_i la variable aléatoire égale à 1 si la boule tirée au rang i est blanche et égale à 0 sinon.
On pose enfin, $X_0 = 0$ et, pour $k \in \mathbb{N}^*$, $Z_k = X_k - X_{k-1}$

1. Quelle est la loi de T_i pour $i \in \mathbb{N}^*$?
   Ecrire une fonction `Tirage(n, b)` permettant de simuler la loi de T_i, `n` et `b` désignant respectivement le nombre de boules noires et le nombre de boules blanches.
2. Quelle est la loi de X_1 ? Ecrire une fonction `G(n, b)` permettant de simuler la loi de X_1
3. On admet que, pour $k \in \mathbf{N}^*$, Z_k suit la même loi que X_1
   (a) Justifier que $X_k = \displaystyle\sum_{j=1}^{k} Z_j$
   (b) Ecrire une fonction `X(k, n, b)` simulant la loi de X_k
   (c) Donner une valeur approchée de $P(X_2 \leqslant 5)$ dans le cas $n = 3$ et $b = 2$