LA RENCONTRE DU LOUP ET DE L'AGNEAU

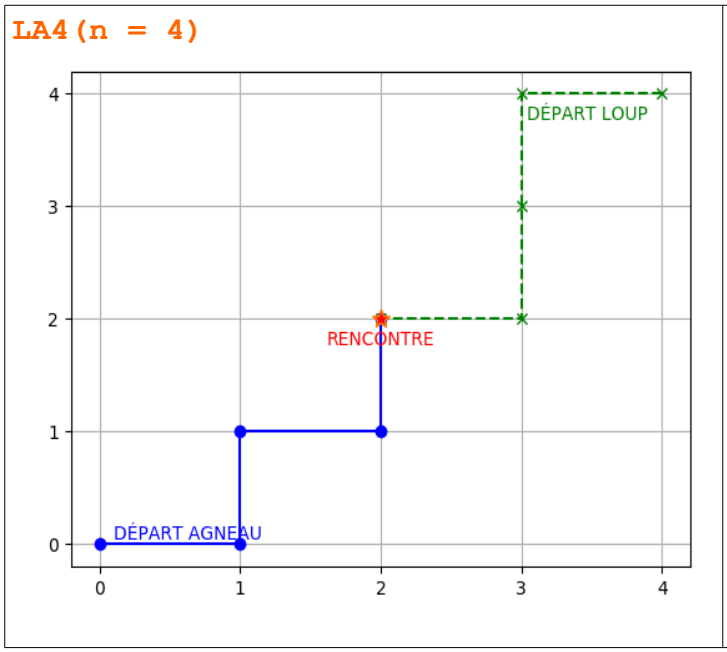
D. Problème classique de marche aléatoire d'après deux articles du bulletin vert de l'APMEP

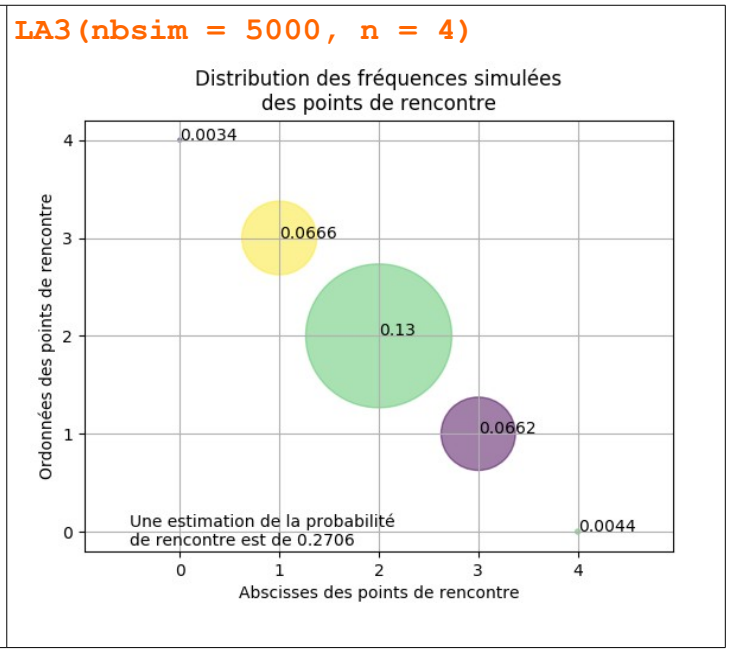
- Deux articles du bulletin vert de l'**APMEP** ont attiré mon attention, pour lesquels je me suis dit qu'ils pouvaient donner lieu à une intéressante séquence de TP d'algorithmique et de leur mise en œuvre en Python. Ce sont le **bulletin n° 515. p. 401-406 – Le loup et l'agneau – Une activité de probabilités en seconde** et le **bulletin n° 516. p. 637-639 – Le loup et l'agneau (cas général)**.
- **L'agneau** est placé au départ en **(0 ; 0)** et le **loup** en **(n ; n)** et ils se déplacent sur les lignes d'un quadrillage **(0 ; n)×(0 ; n)**. Pour se déplacer l'agneau et le loup lancent à tour de rôle une pièce de monnaie à deux faces distinctes équilibrées. Si elle tombe sur **pile l'agneau se déplace de 1 en x ou le loup de –1 en x sinon l'agneau se déplace de 1 en y ou le loup de –1 en y.** Le problème posé dans les articles est celui du calcul de la probabilité qu'ils se rencontrent en **n déplacements au plus**.
- Je vais profiter des possibilités graphiques de l'informatique moderne pour proposer trois objectifs :

  1° Le premier est de simuler **une marche aléatoire de n déplacements** du loup et de l'agneau, d'en faire la représentation graphique dans un repère orthonormé et de repérer s'il y a rencontre ou pas.

  2° Le deuxième sera de simuler nbsim marches aléatoires de n déplacements du loup et de l'agneau et de calculer un fréquence de rencontre, estimation de la probabilité de rencontre.

  3° Le troisième est de faire un cartographie des lieux de rencontre sur le quadrillage (0 ; n)×(0 ; n), avec une représentation graphique de leur fréquence simulée.





D.1. Simulation et représentation graphique de n déplacements d'un point M dans un plan muni d'un repère orthonormé (fichier professeur : LoupAgneau0.py, lignes 9 à 94).

L'algorithme simule une séquence de déplacements d'un point M situé initialement en (0 ; 0), en lançant une pièce équilibrée. Les coordonnées successives des points de déplacement sont enregistrées dans une liste. Le nombre n de déplacements est compté au fur et à mesure. Une représentation graphique peut être faite une fois les déplacements terminés.

D.2. Simulation et représentation graphique d'une marche aléatoire de n déplacements du loup et de l'agneau et repérage d'une éventuelle rencontre (fichier professeur : LoupAgneau0.py, lignes 97 à 217).

L'algorithme simule une séquence d'une marche aléatoire de n déplacements en lançant n fois une pièce équilibrée. Les coordonnées successives des n points de déplacement de l'agneau et du loup sont

enregistrées dans une liste.

● Le test de la rencontre suscite des interrogations : certain (peu nombreux) élèves "voient" de suite que la rencontre ne peut avoir lieu qu'à la fin des n déplacements. D'autres ne voient pas du tout. Dans le doute, le test de rencontre est effectué après chaque déplacement. Quitte à modifier l'algorithme une fois que la démonstration a été faite et comprise que la rencontre ne peut avoir lieu qu'à la fin des n déplacements.

C'est un bon exemple de l'optimisation d'un algorithme. On peut mesure les temps d'exécution (sur la même machine) avec la fonction time.clock() du module time.

Une représentation graphique de la marche est effectuée et l'éventuelle rencontre y est représentée.

D.3. nbsim simulation d'une marche aléatoire de n déplacements pour estimer la probabilité de rencontre (fichier professeur : LoupAgneau0.py, lignes 220 à 257).

L'algorithme effectue nbsim simulations d'une marche aléatoire du loup et de l'agneau et compte le nombre de rencontres. Il affiche la fréquence simulée de rencontre.

Comme il n'y a pas de représentation graphique des marches, je n'enregistre pas les coordonnées de points successifs du déplacement mais simplement les coordonnées du dernier point atteint à la fin de la marche aléatoire.

☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ **PROLONGEMENTS** ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺

D.4. Cartographie des lieux de rencontre et fréquences correspondantes (fichier professeur : LoupAgneau0.py, lignes 260 à 339).

L'algorithme détermine une estimation des lieux de rencontre repérés par leurs coordonnées et affiche la fréquence de rencontre qui y a lieu. Les effectifs et fréquence des points de rencontre sont enregistrés dans un tableau (n+1) × (n+1) qui symbolise le quadrillage.

Nom Prénom :

TP_ALGO_D.1. (Fiche élève 1/2) – Marche aléatoire d'un point M sur le quadrillage d'un plan repéré. Fichier LoupAgneau0_Elev.py lignes 9 à 30.

● Dans Spyder, exécuter les lignes de commandes des **lignes 9 à 30** et compléter le tableau suivant en recopiant les résultats affichés dans la console.

```
"Comment déplacer aléatoirement un point M(x ; y) sur un échiquier et enregistrer ses déplacements ?"
"Je jette un pièce équilibrée. Si elle tombe sur pile je déplace M de 1 en x."
"Si elle tombe sur face je déplace M de 1 en y. Soit M(0 ; 0) la position de départ de M"
"Les positions successives de M sont enregistrées dans deux listes,"
"    une liste posiMX des valeurs successives de x,"
"    une liste posiMY des valeurs successives de y,"
from random import sample

"Création de la pièce, position initiale du point M, valeur initiale de n"
piece = ('pile', 'face')
posiMX = [0] ; posiMY = [0] ; n = 0

"Je sélectionne et j'exécute les lignes 27 à 45 (dans l'éditeur de script les lignes sont numérotées)."
"J'observe le contenu des variables piece, n, posiMX, posiMY, dans la fenêtre ad-hoc."
"Je les note au bon endroit sur le document."
"Je répète cette exécution 10 fois. Tous les élèves auront des résultats différents."

"Je lance la pièce et j'ajoute 1 à n compteur du nombre de déplacements"
tirage = sample(piece, 1) ; n += 1

10 Résultats piece :  0 ;  ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ;  ...


10 Résultats n :      0 ;   1 ;  ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ;  10

"Je code l'issue 'pile' par 1 et l'issue 'face' par 0"
issue = int(tirage[0] == 'pile')

Résultats :                ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ;  ...
```

Nom Prénom :

TP_ALGO_D.1. (Fiche élève 2/2) – Marche aléatoire d'un point M sur le quadrillage d'un plan repéré. Fichier LoupAgneau0_Elev.py lignes 32 à 62.

- Dans Spyder, exécuter les lignes de commandes des **lignes 32 à 62** et compléter le tableau suivant en recopiant les résultats affichés dans la console.
  Le graphique sera copié-collé dans un document texte (Libre office par exemple).

```
"Si l'issue vaut 1 je déplace M de 1 en x en ajoutant 1 à la dernière valeur de x posiMX[-1],"
if (issue == 1):
    nouvelleValeurX = posiMX[-1] + 1
    "Je positionne cette dernière valeur de x à la fin de la liste des valeurs de x"
    posiMX.append(nouvelleValeurX)
    "La nouvelle valeur de y, positionnée en fin de liste, est égale à l'ancienne"
    posiMY.append(posiMY[-1])
else:
    "Sinon c'est la dernière valeur de y que j'augmente de 1."
    nouvelleValeurY = posiMY[-1] + 1
    "Je positionne cette dernière valeur de y à la fin de la liste des valeurs de y"
    posiMY.append(nouvelleValeurY)
    "La nouvelle valeur de x, positionnée en fin de liste, est égale à l'ancienne"
    posiMX.append(posiMX[-1])
10 Résultats posiMX : 0 ;  ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ;  ...

10 Résultats posiMY : 0 ;  ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ;  ...

"Représentation graphique de la marche du point M, après avoir exécuté 10 fois les lignes précédentes"
from matplotlib import pyplot as plt
plt.figure()
plt.plot([0, n], [0, n], ls = 'none')
plt.xticks(range(n + 1)) ; plt.yticks(range(n + 1))
plt.text(0.1, 0, 'DÉPART M', va = 'bottom', ha = 'left', color = 'blue', rotation = 45)
plt.plot(posiMX, posiMY, 'bo-')
plt.plot(posiMX[n], posiMY[n], marker = '*', markerfacecolor = 'red', markersize = 10)
plt.text(posiMX[n], posiMY[n], 'TERMINUS M', va = 'bottom', ha = 'left', color = 'green')
plt.title('MARCHE ALÉATOIRE DU POINT M')
plt.xlabel('Abscisses x') ; plt.ylabel('Ordonnées y')
plt.grid() ; plt.show()
```

Nom Prénom :

TP_ALGO_D.2. (Fiche élève 1/2) – Simulation et représentation graphique d'une marche aléatoire de 10 déplacements du loup et de l'agneau et sur les lignes d'un quadrillage (0 ; 10)×(0 ; 10) et repérage d'une éventuelle rencontre. Fichier LoupAgneau0_Elev.py lignes 98 à 149.

● Dans Spyder, exécuter les **lignes 107 à 149** et compléter le tableau suivant en recopiant les résultats affichés dans la console.

```
"Les n positions de l'agneau et du loup sont enregistrées, au fur et à mesure"
"    des tirages, dans des listes afin de pouvoir illustrer les déplacements"
"L'agneau démarre en (0 ; 0) et le loup en (n ; n) L'agneau joue le premier"
from random import sample

"Nombre n de déplacements prévus sur un quadrillage (0 ; n)×(0 ; n). Création de la pièce"
n = 10
piece = ('pile', 'face')

"La marche aléatoire, L'agneau démarre en (0 ; 0) et le loup en (n ; n)."
"L'indicatrice de rencontre est positionnée à 0."
posisAgnX = [0] ; posisAgnY = [0]
posisLoupX = [n] ; posisLoupY = [n]
rencontre = 0

"n déplacements de l'agneau et du loup"
for i in range(0, n):
    "l'agneau jette une pièce"
    tirAgn = sample(piece, 1)
    "Si elle tombe sur 'pile' il se déplace de 1 en x sinon de 1 en y"
    depAgn = int(tirAgn[0] == 'pile')
    if (depAgn == 1):
        posisAgnX.append(posisAgnX[-1] + 1)
        posisAgnY.append(posisAgnY[-1])
    else:
        posisAgnY.append(posisAgnY[-1] + 1)
        posisAgnX.append(posisAgnX[-1])
    "le loup jette une pièce"
    tirLoup = sample(piece, 1)
    "Si elle tombe sur 'pile' il se déplace de -1 en x sinon de -1 en y"
    depLoup = int(tirLoup[0] == 'pile')
```

Nom Prénom :

```
    if (depLoup == 1):
        posisLoupX.append(posisLoupX[-1] - 1)
        posisLoupY.append(posisLoupY[-1])
    else:
        posisLoupY.append(posisLoupY[-1] - 1)
        posisLoupX.append(posisLoupX[-1])

    "Si les cordonnées du loup et de l'agneau sont égales c'est qu'il y a rencontre"
    if ((posisAgnX[-1] == posisLoupX[-1]) & (posisAgnY[-1] == posisLoupY[-1])):
        rencontre = 1
        print('Rencontre en (', posisAgnX[-1], ";", posisAgnY[-1], ')')
    else:
        print('Pas de rencontre')

10 résultats posisAgnX :    0 ;  ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ;    ...

10 résultats posisAgnY :    0 ;  ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ;    ...

10 résultats posisLoupX :   0 ;  ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ;    ...

10 résultats posisLoupY :   0 ;  ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ;    ...

10 résultats rencontre :    N ;  ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ;    ...
(quand il y a rencontre, on affiche les coordonnées du point de rencontre)

#Si l'on fait le test de rencontre à la fin des n déplacements, comment modifier les 5 dernières lignes de
programme ?
Réponse :
```

Nom Prénom :

TP_ALGO_D.2. (Fiche élève 2/2) – Représentation graphique de la marche aléatoire de 10 déplacements du loup et de l'agneau et repérage d'une éventuelle rencontre. Fichier LoupAgneau0_Elev.py lignes 153 à 169.

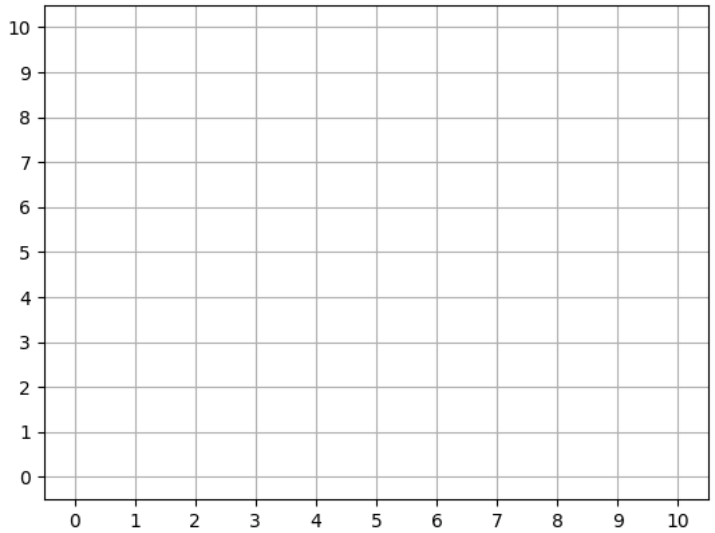
1° Utiliser les résultats de la page 6 et les instructions graphiques ci-dessous pour tracer le graphique des 10 déplacements aléatoires.

2° Vérifier en exécutant les commandes des **lignes 153 à 169**.

```
"Graphique des chemins du loup et de l'agneau"
from matplotlib import pyplot as plt
plt.figure()
plt.plot([0, n], [0, n], ls = 'none')
plt.xticks(range(n + 1)) ; plt.yticks(range(n + 1))
plt.plot(posisAgnX, posisAgnY, 'bo-')
plt.plot(posisLoupX, posisLoupY, 'gx--')
plt.text(0.1, 0, 'DÉPART AGNEAU', va = 'bottom', ha = 'left', color = 'blue', rotation = 45)
plt.text(n - .1, n - .1, 'DÉPART LOUP', va = 'top', ha = 'right', color = 'green', rotation = 45)
if rencontre == 1:
    plt.plot(posisAgnX[n], posisAgnY[n], marker = '*', markerfacecolor = 'red', markersize = 10)
    plt.text(posisAgnX[n], posisAgnY[n] - .1, 'RENCONTRE', va = 'top', ha = 'center', color = 'red')
plt.grid() ; plt.show()
```

- **Prolongement 1 :** reprendre des lignes de **D.1.** pour en faire une fonction MD1.
  Compléter les **lignes 69 à 81** du fichier **LoupAgneau0_Elev.py.**
  Exécuter et utiliser la fonction pour simuler 4 marches aléatoires dont on copiera les représentation graphique dans un fichier texte (Libre office par exemple).

- **Prolongement 2 :** reprendre des lignes de **D.2.** pour en faire une fonction LA4.
  Compléter les **lignes 176 à 201** du fichier **LoupAgneau0_Elev.py.**
  Exécuter et utiliser la fonction pour simuler 4 marches aléatoires dont on copiera les représentation graphique dans un fichier texte (Libre office par exemple).



Nom Prénom :

TP_ALGO_D.3. (Fiche élève 1/2) – nbsim simulation d'une marche aléatoire de n déplacements pour estimer la probabilité de rencontre, en 2 fonctions, marche(…) et rencontres(…). Fichier LoupAgneau0_Elev.py lignes 220 à 248.

● Dans la partie droite du tableau rédiger, en face des lignes de code correspondantes, des commentaires expliquant le fonctionnement des **lignes 230 à 248**.

```
"Je m'inspire de LA4 pour construire une fonction qui simule n déplacements"
"    cad une marche aléatoire,  et indique une éventuelle rencontre."
"Il n'est pas nécessaire de garder la trace de tous les déplacements,"
"    seule la position finale est nécessaire et suffisante"
"Je ne tire plus une pièce mais directement un entier parmi 0 ou 1 équiprobables"
"positAgn et positLoup prennent pour valeurs les coordonnées successives"
"    du loup et de l'agneau au cours de leur marche aléatoire"
"À la fin de la marche je teste l'égalité des coordonnées du loup et de l'agneau"
def marche(n = 4):
    from random import randint
    positAgnX = 0 ; positAgnY = 0
    positLoupX = n ; positLoupY = n
    rencontre = 0
    for i in range(n):
        tirAgn = randint(0, 1)
        if (tirAgn == 1):
            positAgnX += 1
        else:
            positAgnY += 1
        tirLoup = randint(0, 1)
        if (tirLoup == 1):
            positLoupX -= 1
        else:
            positLoupY -= 1
    if ((positAgnX == positLoupX) & (positAgnY == positLoupY)):
        rencontre = 1
    return(rencontre)
```

Nom Prénom :

TP_ALGO_D.3. (Fiche élève 2/2) – nbsim simulation d'une marche aléatoire de n déplacements, pour estimer la probabilité de rencontre, en 2 fonctions, marche(…) et rencontres(…). Fichier LoupAgneau0_Elev.py lignes 250 à 257.

● Dans la partie droite du tableau rédiger, en face des lignes de code correspondantes, les commentaires les expliquant des **lignes 252 à 257.**

```
"J'utilise la fonction marche(...) dans nbsim simulations pour estimer"
"    la probabilité de rencontre. Utilise la fonction marche(...)"
def rencontre(n = 4, nbsim = 5000):

    rencontres = 0

    for i in range(nbsim):

        rencontres += marche(n)

    estimProba = round(rencontres / nbsim, 4)

    print('Une estimation de la probabilité de rencontre est :', estimProba)
```

1° Utiliser ces deux fonctions pour faire 10 estimations de la probabilité de rencontre, avec 1000 simulations :

10 résultats : ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ; ...

2° Utiliser ces deux fonctions pour faire 10 estimations de la probabilité de rencontre, avec 100 000 simulations :

10 résultats : ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ; ... ; ...

3° Comparer ces deux séries de résultats et conclure quant à l'effet du nombre de simulations sur les fréquences de rencontre.

4° Utiliser ces fonctions pour étudier l'effet de la longueur n de la marche sur les fréquences de rencontre . Prendre 5 valeurs de n : 4, 10, 20, 50, 100.

5 résultats : ... ; ... ; ... ; ... ; ...

5° Conclure à propos de l'effet de la longueur de la marche sur les fréquences de rencontre :

Nom Prénom : **PROLONGEMENTS**

TP_ALGO_D.4. (Fiche élève) – Cartographie des lieux de rencontre.

Utiliser les fonctions

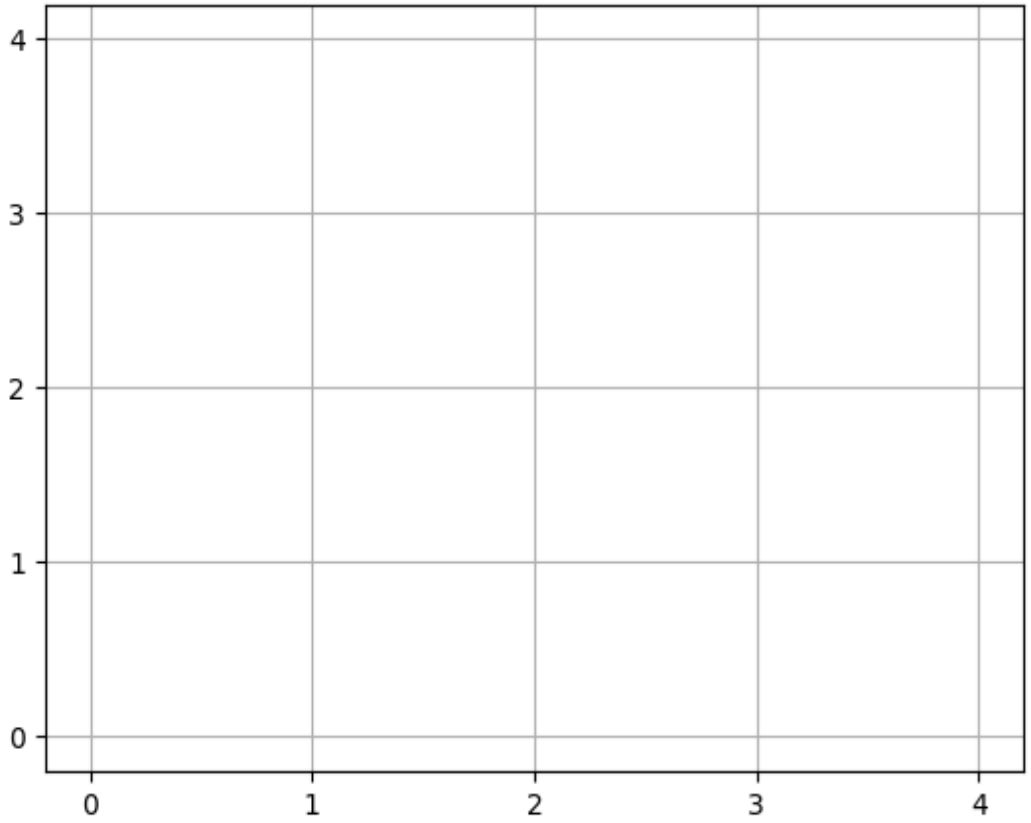
LA1(…) **lignes 265 à 290,**

LA2(…) **lignes 292 à 306,**

LA3(…) **lignes 308 à 339,**

pour simuler les lieux de rencontre de l'agneau et du loup et estimer les fréquences qui leur correspond.

1° Reproduire le résultat d'une des simulations sur le graphique ci-contre, pour une marche de 4 déplacements.

2° Commenter les résultats obtenus.

3° Faire 2 autres simulations et copier les graphiques obtenus dans un fichier texte avec une mise en page au format paysage.