

Plaquette 1 - Seuils, Monte-Carlo et marches aléatoires

Algorithmique et Python — Terminale STI2D

Exercices

Méthode

Trois familles d'algorithmes au programme : la recherche de **seuil** (boucle « tant que »), la méthode de **Monte-Carlo** (estimer une probabilité ou une aire par simulation) et les **marches aléatoires** (position après n pas). On lit d'abord, on écrit ensuite.

Exercice 1 - Dérouler une boucle while

Que renvoie cette fonction ?

```
def f():  
    u = 5000  
    n = 0  
    while u > 2000:  
        u = u * 0.85  
        n = n + 1  
    return n
```

Exercice 2 - Écrire un algorithme de seuil

Une population de bactéries de 2 000 individus croît de 9 % par heure. Écrire une fonction qui renvoie le nombre d'heures au bout duquel elle dépasse 10 000, et donner cette valeur.

Exercice 3 - Simuler une probabilité simple

Compléter le programme qui estime la probabilité d'obtenir un 6 en lançant un dé, puis indiquer la valeur attendue.

```
from random import randint  
  
def frequenceSix(N):  
    succes = 0  
    for i in range(N):  
        de = randint(1, ...)   
        if de == 6:  
            succes = succes + 1  
    return succes / ...
```

Exercice 4 - Simuler une loi binomiale

On veut estimer $P(X \geq 1)$ où $X \sim B(25; 0,03)$ (pièces défectueuses).

Écrire la fonction de simulation et donner la valeur théorique.

Exercice 5 - Monte-Carlo : estimer une aire

On tire au hasard un point dans le carré $[0; 1] \times [0; 1]$ et on compte les points vérifiant $y < x^2$.

- Que représente la fréquence obtenue ?
- Compléter le programme.

```
from random import random

def aire(N):
    dedans = 0
    for i in range(N):
        x = random()
        y = random()
        if y < ...:
            dedans = dedans + 1
    return dedans / N
```

Exercice 6 - Monte-Carlo : estimer π

On tire un point au hasard dans le carré $[0; 1] \times [0; 1]$ et on teste $x^2 + y^2 < 1$.

- Quelle est la probabilité théorique de cet événement ?
- Comment en déduire une estimation de π ?

Exercice 7 - Marche aléatoire : lecture

Que fait ce programme, et quelles valeurs peut-il renvoyer pour $n = 4$?

```
from random import randint

def marche(n):
    position = 0
    for i in range(n):
        if randint(0, 1) == 1:
            position = position + 1
        else:
            position = position - 1
    return position
```

Exercice 8 - Marche aléatoire : loi de la position

Pour la marche aléatoire à 4 pas, calculer la probabilité de terminer en 0, puis en 4.

Exercice 9 - Moyenne de simulations

Écrire une fonction *distanceMoyenne*(n, N) qui simule N marches aléatoires de n pas et renvoie la moyenne des **valeurs absolues** des positions finales.

Exercice 10 - Corriger un programme

Ce programme doit estimer la fréquence de « face » sur N lancers. Il comporte deux erreurs : les trouver et donner la version corrigée.

```
from random import random
```

```
def frequenceFace(N):
    faces = 0
    for i in range(N):
        if random() < 0.5:
            faces = faces + 1
    print(faces)
```

Exercice 11 – Problème : simuler un contrôle de lot

Un lot est accepté si, sur 10 pièces prélevées dans une production à 5 % de défauts, au plus une est défectueuse.

- Écrire une fonction estimant la fréquence d'acceptation sur N simulations.
- Donner la valeur théorique attendue.

Exercice 12 – Une somme d'inverses

On considère la fonction suivante.

```
def somme(n):
    s = 0
    for k in range(1, n + 1):
        s = s + 1 / k
    return s
```

- Écrire avec le symbole Σ la somme calculée par *somme(n)*.
- Calculer *somme(4)* à la main, en valeur exacte puis approchée au millième.
- Quel est le rôle de chacune des variables s et k ?

Exercice 13 – La somme des premiers cubes

- Écrire une fonction Python *cubes(n)* qui renvoie $\sum_{k=1}^n k^3$.
- Que renvoie *cubes(5)* ?
- Vérifier que ce résultat est égal à $\left(\frac{5 \times 6}{2}\right)^2$.

Exercice 14 – Intercaler un point par les moyennes

On considère la fonction suivante, où $A(x_1; y_1)$ et $B(x_2; y_2)$ sont deux points d'ordonnées positives.

```
from math import sqrt
def milieu(x1, y1, x2, y2):
    return (x1 + x2) / 2, sqrt(y1 * y2)
```

- Que renvoie *milieu(0, 100, 4, 400)* ?
- On sait que A et B sont sur la courbe de $f(x) = 100 \times \sqrt{2}^x$. Le point obtenu est-il sur cette courbe ?

Exercice 15 - Synthèse : seuil sur une simulation

On veut savoir combien de lancers d'un dé sont nécessaires, en moyenne, pour obtenir un premier 6.

- a) Écrire une fonction *premierSix()* qui renvoie le nombre de lancers nécessaires pour une simulation.
- b) Écrire une fonction qui renvoie la moyenne sur N simulations.
- c) Quelle valeur théorique doit-on retrouver ?