

```

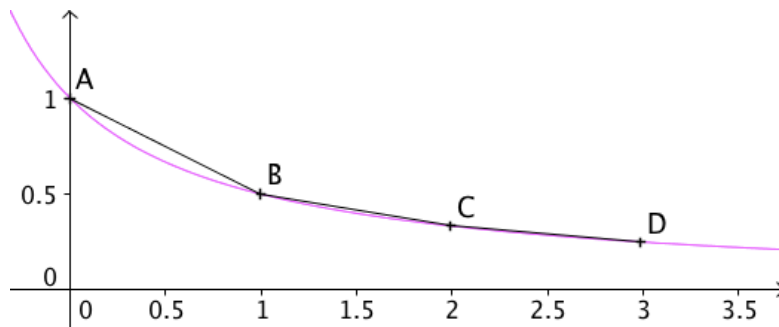
s = experience()
n = 1
L = [s] # moyenne su
while n < nExperiences:
    n = n+1
    s = s + experience
    L.append(s/n) # or
plt.plot(list(range(1,
plt.plot([1, nExperiences

```

LONGUEUR D'UNE COURBE

Dans un repère orthonormé, on veut calculer, sur l'intervalle $[0 ; 3]$, une valeur approchée de la longueur de la courbe de la fonction f définie par $f(x) = \frac{1}{x+1}$.

1) Pour cela, on a placé sur la courbe quatre points A, B, C et D d'abscisses respectives 0, 1, 2 et 3 formant trois segments [AB], [BC] et [CD]. En calculant la somme $AB + BC + CD$ donner une première approximation de la longueur de la courbe de la fonction f sur l'intervalle $[0 ; 3]$.



2) Une meilleure approximation s'obtient avec un plus grand nombre de points sur la courbe dont les abscisses sont réparties régulièrement sur l'intervalle $[0 ; 3]$. L'algorithme à compléter suivant permet d'obtenir une approximation de la longueur de la courbe de la fonction f sur l'intervalle $[0 ; 3]$ en fonction du nombre N de segments ainsi formés.

```

L ← 0
p ← 3/N
x1 ← 0
x2 ← x1 + p
Pour i allant de 1 à N
    y1 ← 1/(x1 + 1)
    y2 ← 1/(x2 + 1)
    L ← L + ...
    x1 ← x1 + p
    x2 ← x2 + p
Fin Pour
Afficher L

```

- 1) a) Que permet de calculer la variable p .
b) Compléter la ligne 8 de l'algorithme.
- 2) Programmer et tester l'algorithme précédent pour différentes valeurs de N .
Avec Python : La syntaxe pour "racine carrée" est `sqrt`.
Saisir au début du programme `from math import*`
- 3) Adapter le programme pour obtenir une approximation de la longueur de la courbe de la fonction f sur l'intervalle $[1 ; 5]$. Donner cette longueur.



Hors du cadre de la classe, aucune reproduction, même partielle, autres que celles prévues à l'article L 122-5 du code de la propriété intellectuelle, ne peut être faite de ce site sans l'autorisation expresse de l'auteur.

www.maths-et-tiques.fr/index.php/mentions-legales