Le calcul d'une racine carrée

La méthode de Héron

La suite définie par :

$$\begin{cases} x_0,\ a>0 \\ x_{n+1}=\dfrac{1}{2}\left(x_n+\dfrac{a}{x_n}\right) \end{cases}$$

Converge vers $\sqrt{a}$ assez rapidement pour peu que l'on choisisse x_0 assez proche de la valeur de $\sqrt{a}$

Exemple de calcul : $\sqrt{2}$ avec $x_0 = 1$

$$x_1 = \frac{1+2}{2} = \frac{3}{2} = 1{,}5,$$
$$x_2 = \frac{3/2+4/3}{2} = \frac{17}{12} = 1{,}4166\ldots,$$
$$x_3 = \frac{17/12+24/17}{2} = \frac{577}{408} = 1{,}4142156\ldots,$$
$$x_4 = \frac{577/408+816/577}{2} = \frac{665857}{470832} = 1{,}4142135623745\ldots,$$
$$x_5 = \frac{886731088897}{627013566048} = 1{,}41421356237309504880168 96\ldots,$$

Nous allons concevoir deux algorithmes :

- Un premier algorithme pour trouver une valeur proche de $\sqrt{a}$,
- Un second algorithme pour calculer $\sqrt{a}$ avec la méthode de Héron.

Le premier algorithme

On va appliquer une recherche par dichotomie:
Le nombre cherché est dans l'intervalle $[0,a]$
Puis il sera soit dans l'intervalle $\left[0,\dfrac{0+a}{2}\right]$ soit dans l'intervalle $\left[\dfrac{0+a}{2},a\right]$ et ainsi de suite.
On s'arrêtera lorsque l'intervalle aura une longueur inférieure à 1

? Exercice 1 :

Écrire l'algorithme

.....
.....
.....
.....
.....
.....
.....
.....
.....

Exercice 2 :

Compléter fonction `racine1(a)`, qui prend en paramètre un nombre positif a et qui retourne une valeur proche de sa racine carrée, en utilisant l'algorithme précédent.

```
def racine1(a):
'''spécifications


'''
    inf = 0
    sup = a
    while ..................:
        mid = (inf + sup)/2
        if mid*mid..............:
            ..................
        else:
            ..................
    return sup
```

Question 1:

Que pouvez vous dire de la terminaison de ce programme en considérant le variant de boucle : $sup - inf$?

.....

.....

.....

.....

.....

Le second algorithme

Exercice 3 :

Écrire une fonction `racine2(a)` qui prend en paramètre un nombre positif a , et qui retourne le 10 ème terme de la suite x_n de la méthode de Héron en ayant initialisé le terme x_0 avec le résultat de la fonction `racine1(a)`.

```
def racine2(a):




-
```

Tester le programme sur différents nombres