

Recherche de la racine carré d'un nombre

Téléchargez le fichier « `racine_carre.py` », à compléter au fur et à mesure des exercices.

I. Recherche par dichotomie

L'algorithme de recherche par dichotomie peut être utilisé pour trouver la racine carré d'un nombre à ϵ près. **Bien évidemment, il est interdit d'utiliser la fonction `sqrt()` du module `math`.**

1. On cherche dans un premier temps les deux entiers x_1 et x_2 dont le carré encadrent le nombre. Des deux entiers nous permettront d'initialiser le début et la fin de l'encadrement de recherche.
2. Dans un second temps, on utilise un algorithme de recherche dichotomique pour trouver la valeur de x telle que x^2 soit égale au nombre, à ϵ près.

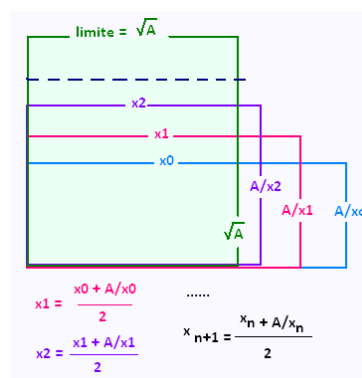
Implémentez cette méthode dans une fonction `racine_dichotomie()` prenant *nombre* et *epsilon* en argument, et renvoyant la racine carré de *nombre*.

II. Méthode de Héron

Cette méthode déjà connue par les Babyloniens, est également attribuée au grec Héron d'Alexandrie (1^{er} siècle). Il l'expose dans le premier tome de son ouvrage *Metrica*, ouvrage qui a été découvert en 1896. Chez les mathématiciens grecs, extraire la racine carrée de a revient à trouver un carré dont l'aire soit égale à a . En prenant un rectangle de côté arbitraire $L_0 = x$ et de même aire, il est nécessaire que l'autre côté ait pour longueur $l_0 = \frac{a}{x}$. Pour le rendre « moins rectangle », il suffit de considérer un nouveau rectangle dont les dimensions vérifient :

- ▷ la longueur est la moyenne arithmétique des dimensions du rectangle précédent ;
- ▷ pour que l'aire reste égale à a , on divise a par cette nouvelle longueur pour trouver l'autre dimension du rectangle.

En répétant infiniment le processus, le rectangle se transforme petit à petit en un carré de même aire. Cette constatation est à la base de la méthode de Héron.



Implémentez cette méthode dans une fonction `racine_heron()` prenant *nombre* et *epsilon* en argument, et renvoyant la racine carré de *nombre*.

III. Comparaison de l'efficacité en temps

Pour mesurer la durée d'exécution d'une instruction, on peut utiliser le module `timeit`.

1. Dans la console Spyder, importez le module `timeit` en saisissant l'instruction :
`from timeit import timeit.`
2. Mesurez le temps d'exécution de la fonction `racine_dichotomie` en saisissant l'instruction :
`timeit(lambda: racine_dichotomie(4, 1e-3)).`
3. Mesurez le temps d'exécution de la fonction `racine_heron` en saisissant l'instruction :
`timeit(lambda: racine_heron(4, 1e-3)).`
4. Quelle méthode vous paraît la plus efficace ?

IV. Sources

▷ <http://ww2.ac-poitiers.fr/math/spip.php?article1087>