

0.1 Méthode de Lindenmayer ou L-systèmes

Inventée par le biologiste hongrois Lindenmayer en 1968, cette méthode était destinée à modéliser la croissance des plantes ou des bactéries. A partir d'un mot initial et d'opérations de remplacement, on dessine des figures. On considère l'alphabet $\{B, g, d, A\}$. Le k -ième itéré du L-système s'obtient en appliquant k fois la substitution au mot de départ.

Exemple: A partir du mot initial **BgAdB** et de la règle de substitution **A** → **ABA**, on obtient **BgABAdB** après une itération. Ecrire le résultat de la deuxième itération.

On peut appliquer plusieurs règles à la fois, par exemple la règle de substitution: **B** → **BB** en plus de la précédente, les règles s'appliquent en même temps.

0.1.1 A faire

1. Programme une fonction `remplacerL(mot, lettre, motif)` qui remplace une lettre par un motif dans un mot.

```
>>>mot = "AdAAg"
>>>remplacerL(mot, "A", "Ag")
"AgdAgAgg"
```

Chaque lettre A a été remplacée par le motif Ag.

2. Programme une fonction `itererL(depart, regle, k)` qui renvoie le k -ième terme itéré du L-système. `depart` est le mot de départ, la règle de substitution s'écrira sous la forme (lettre, séquence_replacement).

Exemple:

```
>>>itererL('Agd', ('A', 'ABA'), 2)
'ABABABAgd'
```

3. A partir d'un mot donné comme AgAdAAdAdA dans lequel chaque lettre est lue de gauche à droite, on fait correspondre une instruction pour la tortue Python.

A ou B : avance d'une quantité fixée (en traçant),

g : tourne à gauche, sans avancer, d'un angle fixé (le plus souvent 90 degrés),

d : tourne à droite d'un angle fixé.

Les éventuels autres caractères ne font pour le moment rien.

Ecrire une fonction `trace_L(mot, angle=90, echelle=1)` qui affiche le dessin correspondant aux lettres de `mot`. Par défaut l'angle est de 90 degrés.

4. Tester les combinaisons de mots et de règles suivants:
 - (a) **Flocon de Von Koch:** depart : A et regle : $A \rightarrow AgAdAdAgA$
 - (b) ("A", "AdAgAgAAdAdAgA")
 - (c) ("A", "AgAAdAAdAdAgAgAAdAdAgAgAAgAAAdA")
 - (d) ("A", "AAdAdAdAdAA")
 - (e) ("A", "AAdAddAdA")
 - (f) ("A", "AAdAdAdAdAdAgA")
 - (g) ("A", "AAdAgAdAdAA")
 - (h) ("A", "AdAAddAdA")
 - (i) ("A", "AdAgAdAdA")

0.1.2 Autres tracés

1. Améliorer la fonction d'etracer pour que la tortue avance sans tracer si le caractère rencontré est 'a'. Tester avec la combinaison:
 - depart = "AdAdAdA"
 - regle = ("A", "AgaDag")
2. On veut maintenant autoriser les retours en arrière. Pour cela on ajoute à notre alphabet les symboles [et].
 - Si on rencontre un crochet ouvrant [, on mémorise la position de la tortue sur l'écran et on exécute normalement la séquence entre les crochets.
 - On repart alors de la position mémorisée auparavant et on continue le tracé.

Exemple 1:AgA[AgAdA]gA

1. On avance, on tourne à gauche, on avance.
2. On mémorise la position et on exécute la séquence AgAdA
3. On revient à la position mémorisé et on tourne à gauche, on avance.

Exemple 2:AgA[AgA[dA]gA]gA

1. On avance, on tourne à gauche, on avance.
 - On mémorise la position 1 et on exécute la séquence AgA
 - On mémorise la position 2 et on exécute la séquence dA
 - On revient en position 2 et on exécute gA
2. On revient à la position 1 mémorisée et on tourne à gauche, on avance.