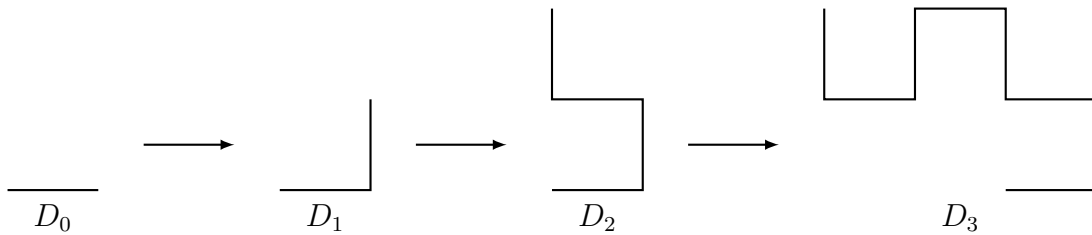


**Exercice 1**

On reprend les fractales : écrire une fonction `dragon(n)` qui dessine la  $n$ -ème itération du dragon de Heighway. L'itération du dragon est la suivante :

- au rang  $n = 0$ , le dragon est un segment;
- au rang  $n + 1$  :
  1. on dessine un dragon de rang  $n$ ;
  2. on tourne de  $90^\circ$ ;
  3. on dessine un dragon de rang  $n$  symétrique.

**Exercice 2**

Python est un langage qui ne gère pas bien la récursivité : par défaut, il empêche d'aller loin dans la récursivité.

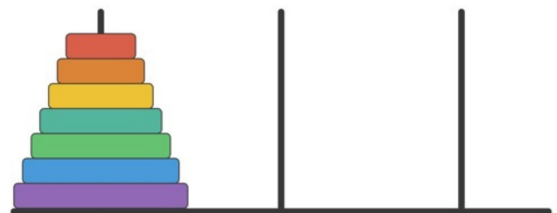
1. Écrire une version récursive de la factorielle.
2. Tester `fact(900)` et `fact(1000)` : lire attentivement le message d'erreur.
3. Par défaut, Python a une profondeur maximale de récursivité. Modifier la profondeur à l'aide des commandes suivantes, et tester :

```
import sys
sys.setrecursionlimit(1500)
```

Attention, cette solution est pratique, mais si vous en avez besoin, c'est que votre code ne devrait peut-être pas être récursif.

**Exercice 3 : tours de Hanoï**

Le problème des tours de Hanoï consiste à résoudre le jeu suivant : vous avez trois pics, les deux à droite vides. Sur le premier sont empilés des disques du plus large au plus petit. Vous ne pouvez déplacer les disques qu'un par un, et uniquement poser un disque sur un disque plus grand, ou sur un pic vide.



Votre objectif est de construire un programme qui représente graphiquement la solution des tours de Hanoï.

Vous êtes laissé·e·s en autonomie pour cette partie du TP. Des conseils pour organiser votre travail :

- Comment comptez-vous représenter en machine une étape de la solution ?
- Comment afficher un pic ?
- Comment afficher 3 pics ?
- Comment afficher les étapes successives de la solution ?
- Et la récursivité, dans tout cela ?

Quelques conseils :

- Parfois, il vaut mieux qu'une fonction renvoie un string plutôt qu'elle n'affiche directement à l'écran.
- Dans un string, les caractères `\n` représentent un saut de ligne, qui est affiché lors d'un `print` : vous pouvez tester `print("bonjour\n coucou")`.
- Lors de votre récursivité, il faut penser à comment modifier la solution de rang  $n - 1$  de manière appropriée.

```

0|0      |      |
00|00    |      |
000|000   |      |

00|00    |      |
000|000   |      |
0|0      |      |

000|000   |00|00  |0|0
0|0      |0|0    |0|0
000|000   |00|00  |

0|0      |0|0    |
000|000   |00|00  |000|000

0|0      |00|00  |000|000
0|0      |00|00  |000|000

0|0      |      |00|00
0|0      |      |000|000

0|0      |      |0|0
000|000   |      |00|00
000|000   |      |000|000

```