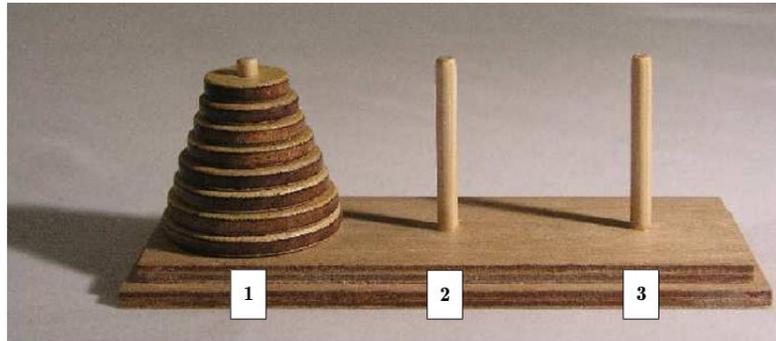


TP5 – TOUR DE HANOI

OBJECTIFS :

Les Tours de Hanoï est un jeu inventé par le mathématicien Édouard Lucas en 1883. Il est constitué de trois piquets verticaux, notés 1, 2 et 3 et de n disques superposés de tailles strictement décroissantes avec un trou au centre et enfilés autour du piquet 1.

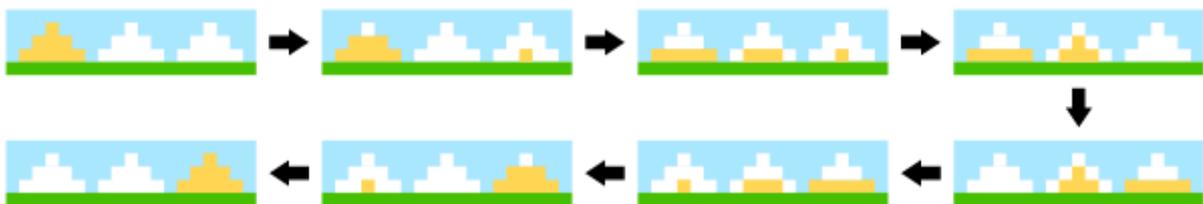


Le but du jeu consiste à déplacer l'ensemble des disques pour que ceux-ci se retrouvent enfilés autour du piquet 3 en respectant les règles suivantes :

- les disques sont déplacés un par un ;
- un disque ne doit pas se retrouver au-dessus d'un disque plus petit. (On suppose évidemment que cette dernière règle est également respectée dans la configuration de départ).

Ce problème se résout facilement de manière récursive. Pour déplacer nb disques de la position pos_init à la position pos_fin il faut :

- déplacer les $nb-1$ premiers éléments vers la position restante pos_inter ,
- déplacer l'élément restant (le plus gros) de la position pos_init à la position pos_fin ,
- déplacer la pile des $nb-1$ éléments de la position pos_inter à la position pos_fin sur le plus gros déjà en place.



1. PRELIMINAIRES

Q1. Donner les déplacements nécessaires pour résoudre le problème avec 1, 2 et 3 disques. On pourra abrégé le « Je déplace un disque du piquet 3 au piquet 2 » par « 3→2 ».

Q2. Déterminer la formule permettant de calculer la position intermédiaire pos_inter en fonction de pos_init et pos_fin .

Q3. Écrire sous Python une fonction récursive `hanoi(nb, pos_init, pos_fin)` résolvant le problème en affichant les étapes sous la forme « Déplacer le disque de 1 vers 0 ».

« Interface graphique »

Ouvrir le programme « *hanoi_graph.py* » permettant de visualiser les coups.

Q4. Compléter ce programme en ajoutant la fonction trouvée précédemment.

Q5. Améliorer le programme en demandant à l'utilisateur de rentrer les paramètres de la fonction de Hanoi et qu'elle s'applique automatiquement (rappel : on utilisera **input**)

2. COMPLEXITE

Q6. Modifier le programme afin de compter le nombre de coups.

2.1 Estimation de la complexité temporelle

Q7. Mise en place de balises temporelles du module `time`

A l'aide du module `time`, mettre en place un dispositif permettant d'estimer le temps utilisé par la fonction globale, en fonction du nombre de disques. (`import pylab` et `matplotlib`)

Q8. Evaluer indépendamment chacune des fonctions en mettant le temps écoulé dans une liste de 3 valeurs.

Plus joli !

Copier coller ce programme dans la fonction `hanoi_graph` et modifier la fonction `afficher` afin d'afficher maintenant des vrais rectangles qui se déplacent de piquets et piquets.

```
h=1/n
pl.axis([0,1,0,1])
C=['b', 'g', 'r', 'c', 'm', 'y', 'k']

def rectangle(i,d,k):
    pl.plot([(k+1)/4-0.5*(0.25-(n-d)/(5*n)), (k+1)/4+0.5*(0.25-(n-
d)/(5*n)), (k+1)/4+0.5*(0.25-(n-d)/(5*n)), (k+1)/4-0.5*(0.25-(n-
d)/(5*n)), (k+1)/4-0.5*(0.25-(n-
d)/(5*n))], [h*i,h*i,h*(i+1),h*(i+1),h*i], C[d%len(C)], linewidth=3)

def piquet(k,L):
    for i in range(len(L)):
        rectangle(i,L[i],k)
```

