

Projet Python : Les tours de Hanoï

-----| El behedy Ramy |-----

Résolution du problème :

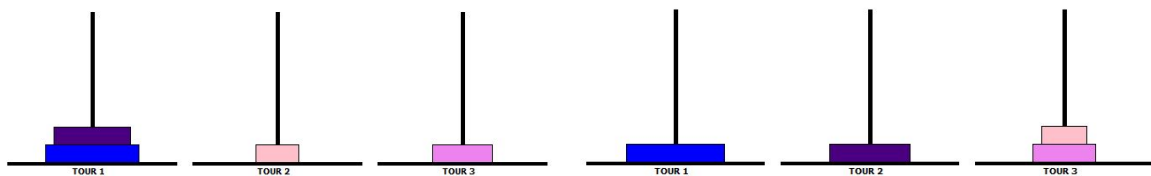
La résolution du problème consiste à empiler sur la tour la plus à droite, les quatre disques triés par ordre croissant sur la tour de gauche.

Pour cela il faudra impérativement utiliser la tige du milieu afin d'avoir la possibilité de déplacer les disques.

4 disques :

taille 1 : beige | taille 2 : rose | taille 3 : violet | taille 4 : bleu

La solution se résout en 15 mouvements, elle est décrite 2 par 2 avec les images ci-dessous :

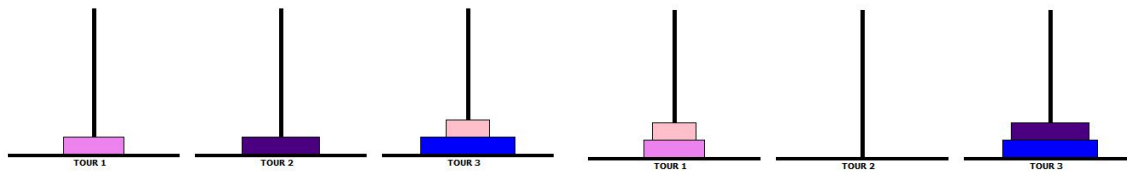


La première étape consiste à déplacer les deux premiers disques à droite et placer le disque violet sur la tige du milieu. On décompose donc en plaçant le plus petit sur le milieu, le rose sur la droite, on replace le plus petit sur le rose et on termine avec le violet au milieu.



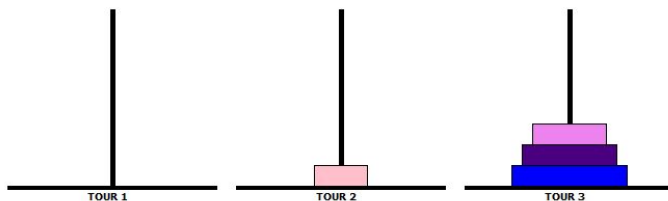
La seconde étape consiste à déplacer le plus gros disque (bleu) à droite afin d'avoir la base de la tour. Pour cela il faudra libérer l'espace de la tour en mettant le plus petit disque (beige) sur le plus gros disque, remplacer le disque

rose sur le disque violet au milieu. Afin de déplacer le disque bleu on met le beige sur le rose et le bleu sur la tour de droite.



Pour la troisième étape il faut placer le disque violet sur le bleu.

On décompose, le beige sur le bleu, le rose sur la tour libre à gauche et enfin le beige se replace sur le rose pour que le violet puisse prendre sa place au dessus du bleu.



Dernière étape, on décompose une dernière fois, le beige au milieu, le rose sur le violet et le beige vient se déposer sur le rose pour compléter la tour.

La fonction hanoi(n, src, des, tmp):

```
def hanoi(n,src,des,tmp):  
    if n == 1:  
        print('Mouvement disque 1 : de la tige {} a la tige  
{}}.'.format(src, des))  
        return  
  
    #On rappelle hanoi en inversant des et tmp  
    hanoi(n - 1, src, tmp, des)  
    print('Mouvement disque {} : de la tige {} a la tige  
{}}.'.format(n, src, des))  
    #On rappelle hanoi en inversant tmp et src  
    hanoi(n - 1, tmp, des, src)
```

Exemple avec 2 disques :

1. La fonction saute le cas ou $n == 1$.
2. Appelle hanoi avec $n-1$ en inversant la tige dest et tmp.
3. La fonction rentre dans la condition $n==1$ et déplace le disque 1 de la tige source à la tige dest (qui est devenu la tige temporaire). Donc le disque 1 se retrouve sur la tige temporaire.
4. On sort de la fonction et la destination redevient des.
5. On déplace le disque 2 de la tige source à la tige tige destination.
6. Appelle hanoi avec $n-1$ (pour remplacer le disque 1) en inversant tmp et des.
7. La fonction rentre dans la condition $n==1$ et déplace le disque 1 de la tige source (qui est devenu la tige temporaire) à la tige des. Donc le disque 1 était sur la tige temp et se retrouve sur le destination.

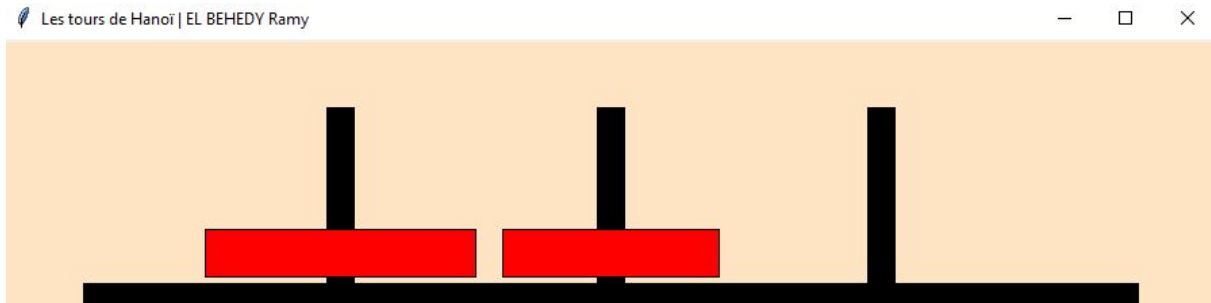
La fonction hanoi(ids, source, destination, temp, hauteurs):

```
def hanoi(ids, source, destination, temp, hauteurs):  
    if len(ids) == 1:  
        time.sleep(0.3)  
  
        # Deplacement et mise a jour des hauteurs  
        move(ids[0], (source, etage(hauteurs[source]-1)), (destination,  
etage(hauteurs[destination])))  
        hauteurs[source] -= 1  
        hauteurs[destination] += 1  
        root.update()  
        return  
  
    #On rappelle hanoi en inversant des et tmp  
    hanoi(copieWithoutLast(ids), source, temp, destination, hauteurs)  
  
    # Deplacement, mise a jour des hauteurs  
    time.sleep(0.3)  
    move(ids[len(ids)-1], (source,  
etage(hauteurs[source]-1)), (destination,  
etage(hauteurs[destination])))  
    hauteurs[source] -= 1  
    hauteurs[destination] += 1  
    root.update()  
  
    #On rappelle hanoi en inversant tmp et src  
    hanoi(copieWithoutLast(ids), temp, destination, source, hauteurs)
```

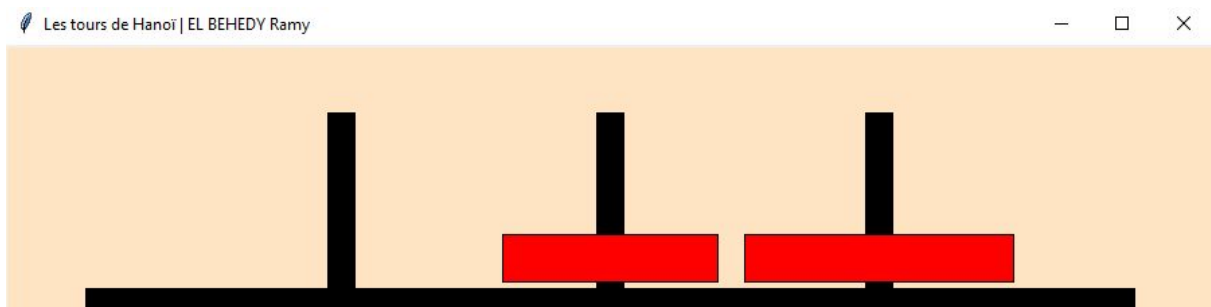
Exemple avec 2 disques :

1. La fonction saute le cas ou la longueurDeLaListe == 1.
2. Appelle hanoi en inversant la tige destination et tmp et avec la fonction copieWithoutLast() qui prend une liste et renvoie un autre liste sans le dernier élément.
3. La fonction rentre dans la condition longueur de la liste == 1 et déplace le disque 1 de la tige source à la tige dest (qui est devenu la

tige temporaire). Le disque 1 se retrouve sur la tige temporaire et les hauteurs sont mises à jours. (la fonction etage converti un nombre en nombre de pixel)



4. On sort de la fonction et la destination redevient la destination de base.
5. On déplace le disque 2 du tige source au tige destination comme pour le premier déplacement.



6. Appelle hanoi en inversant la tige source et tmp et avec la fonction copieWithoutLast().
7. La fonction rentre dans la condition longueurDeLaListe == 1 et déplace le disque 1 de la tige temp à la tige destination.

