

Les tours de Hanoï

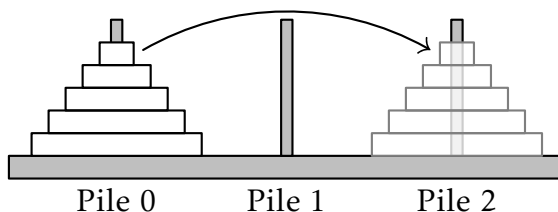
Nom :

Prénom :

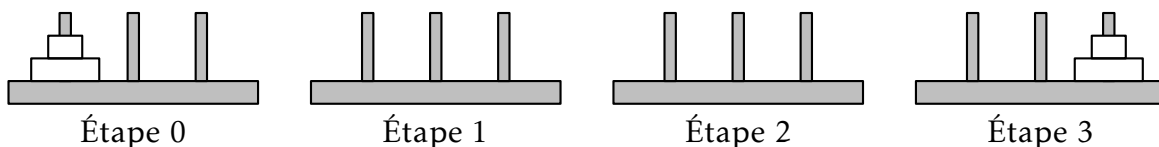
*Les tours de Hanoï*

On appelle **tours de Hanoï** un jeu, inventé par le mathématicien Édouard Lucas, composé de disques de bois s'empilant sur 3 tiges. L'objectif est de déplacer vers la droite la pile de disques se trouvant à gauche, et cela en un minimum de coups. Pour cela il faut respecter les règles suivantes :

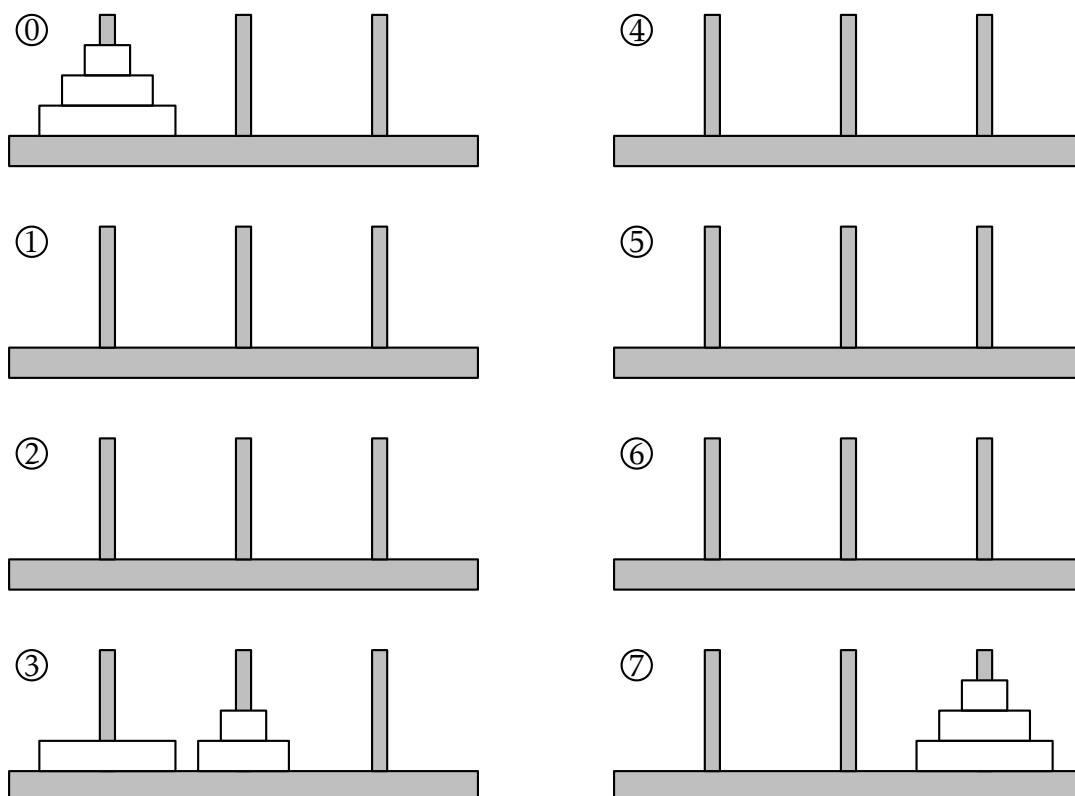
- On ne peut déplacer qu'un disque à la fois.
- On ne peut déplacer qu'un disque sur un autre disque plus grand que lui ou sur un emplacement vide.
- Dans la position initiale, l'empilement de disques respecte la règle précédente.



**EXERCICE 1 :** (1pt) Il faut 3 étapes pour déplacer une pile de 2 disques. Compléter les étapes intermédiaires sur le schéma ci-dessous :



**EXERCICE 2 :** (2,5pt) Il faut 7 étapes pour déplacer une pile de 3 disques. Compléter les étapes manquantes sur le schéma ci-contre :

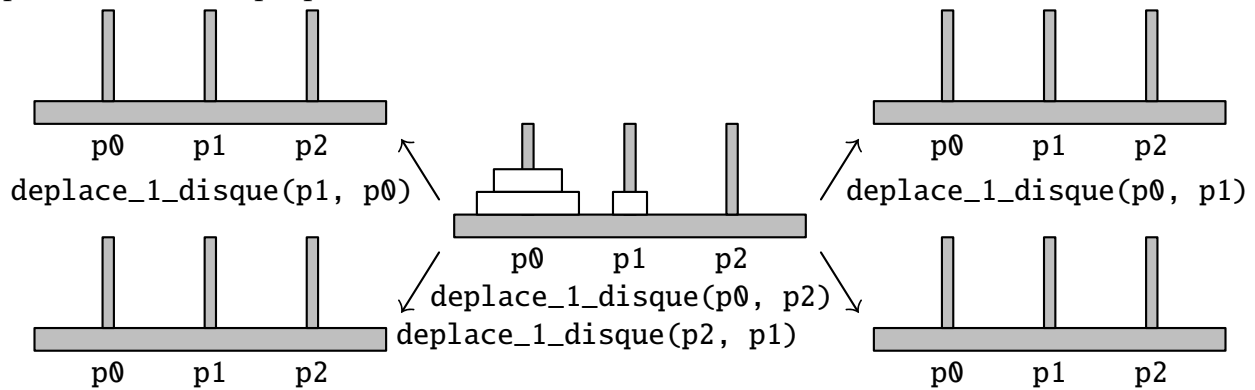


## Résolution des tours de Hanoï

Le but de cette partie est d'étudier la façon de résoudre ce jeu dans le cas général afin de pouvoir ensuite programmer cette méthode. On note  $p_0$ ,  $p_1$  et  $p_2$  les piles sur les 3 tiges, de gauche à droite.

**EXERCICE 3 :** (2pt) On souhaite écrire une fonction `deplace_1_disque` qui prend en paramètres 2 piles origine et cible, et qui déplace la valeur au sommet de origine vers le sommet cible. Si le déplacement n'est pas possible, parce qu'il ne respecte pas les règles du jeu, les piles ne sont pas modifiées.

Dans chacun des cas ci-dessous, en partant de l'état central, dessiner l'état des piles attendu après l'instruction proposée.



**EXERCICE 4 :** (1,5pt) Donner la suite d'instructions nécessaires pour passer de l'étape 0 à l'étape 3 dans l'exercice 1, en utilisant la fonction `deplace_1_disque`.

```
deplace_1_disque(....., .....)
deplace_1_disque(....., .....)
deplace_1_disque(....., .....)

```

**EXERCICE 5 :** (2pt) On veut écrire une fonction récursive `deplace_n_disques` qui prend en paramètres un entier  $n$  et 3 piles origine, cible et stockage, et qui permet de déplacer les  $n$  premiers disques au sommet origine vers cible, en utilisant éventuellement stockage comme pile intermédiaire pour les déplacements.

1) Indiquer l'instruction à utiliser dans le cas  $n == 1$ .

2) On considère maintenant une pile de  $n$  disques sur la pile  $p_0$  qu'on souhaite amener en  $p_2$ , avec  $n > 1$ . En utilisant `deplace_n_disques(n-1, ..., ..., ...)` pour déplacer les  $n - 1$  premiers disques et `deplace_1_disque` pour le dernier disque, donner les instructions permettant d'effectuer les étapes ci-dessous :

