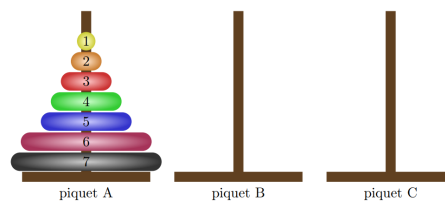


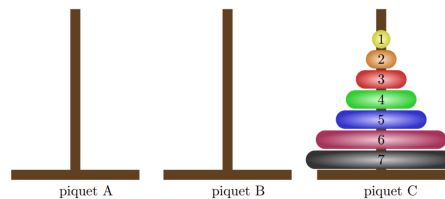
Exercice à faire sur feuille

Tour de Hanoi

Nous allons étudier un jeu et programmer un algorithme pour le résoudre. Ce jeu consiste en trois piquets sur lequel on peut placer des disques de diamètres différents. Comme sur l'image ci-dessous :



Le but du jeu est de déplacer tous les disques du piquet de départ, ci-dessus le piquet A, vers le piquet d'arrivée ci-dessous le piquet C.

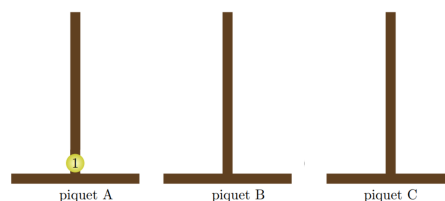


Le joueur peut (et doit) se servir du piquet intermédiaire B pour déplacer les disques car il doit respecter les contraintes de déplacement suivantes :

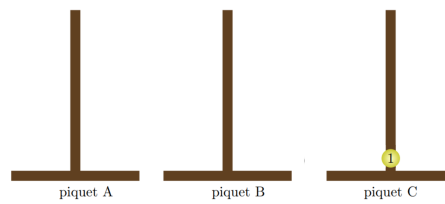
- un seul disque peut être déplacé à la fois,
- Un disque ne doit jamais être posé sur un disque de diamètre inférieur.

Pour se familiariser avec ce jeu, on peut commencer à proposer des solutions à la main pour un petit nombre de disque.

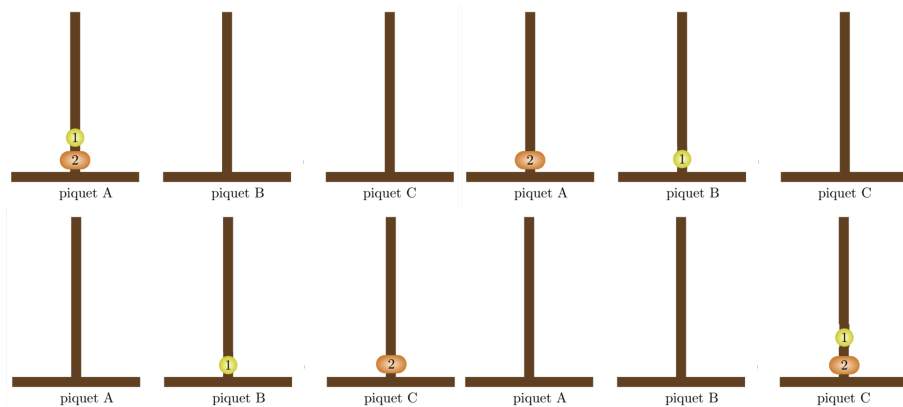
Avec un seul disque on part de la situation initiale suivante :



La solution est alors évidente, il suffit de déplacer le disque sur le piquet C.



Avec deux disques on doit passer par le piquet intermédiaire :



1. A l'aide de schéma successifs des positions des disques sur les trois piquets comme ci-dessous, dessiner toutes les étapes nécessaire pour une situation avec 3 disques.

Nous allons maintenant étudier les outils que nous allons utiliser pour concevoir l'algorithme.

2. Pour représenter les trois piquets, quelle est la structure de données appropriée ? Justifier votre réponse.
3. Quel objet en python permet d'implémenter cette structure de données ?
4. Quelles sont alors les opérations possibles sur cette structure de données ?
5. Quelle fonction permet de les implémenter en python ?
6. Écrire une suite de ligne de commande qui définit la situation initiale du jeu avec un disque.
7. Écrire une suite de ligne de commande qui permet d'effectuer le déplacement voulu entre la situation initiale et finale du jeu à un disque.
8. Au vu des trois exemples fait à la main avec 1 puis 2 puis 3 disques conjecturer comment évolue le nombre de déplacement avec le nombre de disque. Évaluer cette conjecture pour 7 disques, et en déduire la nécessité d'utiliser un algorithme.
9. A l'aide de la question précédente conjecturer la complexité temporelle attendue pour l'algorithme qui résout le jeu.
10. Quelle sera la complexité spatiale de cet algorithme ?
11. Discuter de l'intérêt d'utiliser un programme itératif ou récursif.

Nous allons d'abord choisir d'utiliser un algorithme récursif.

12. Écrire une fonction `deplacement(i,j)` qui déplace un disque du piquet `i` vers le piquet `j`.

L'appel récursif de l'algorithme prend comme argument le nombre de disque à disque restant qui ne sont pas encore bien positionner sur le piquet d'arrivée.

13. Quel sera le cas d'arrêt de l'algorithme récursif?
14. Écrire la partie de l'algorithme correspondant au cas d'arrêt.
15. Justifier de la terminaison de l'algorithme.

Le principe de l'appel récursif peut être compris simplement en observant les déplacements à effectuer pour les cas à 2 disques et le cas à 1 disque.

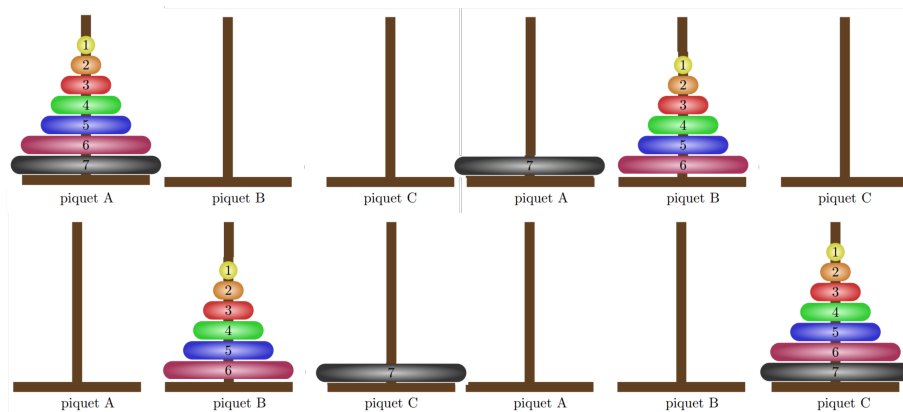
On peut décomposer le cas à 2 disques sur le cas à 1 disque comme ci-dessous :

- on déplace le premier disque sur le piquet intermédiaire
- on déplace le dernier disque sur le piquet d'arrivée
- le piquet intermédiaire devient le nouveau piquet de départ, et on déplace le disque du piquet intermédiaire au piquet d'arrivée.

De même dans le cas à n disque on peut faire appel seulement aux cas à $n-1$ et à 1 disque avec les trois étapes ci-dessous :

- on déplace les $n-1$ premiers disques sur le piquet intermédiaire
- on déplace le dernier disque sur le piquet d'arrivée
- le piquet de intermédiaire devient le nouveau piquet de départ, et on déplace les $n-1$ disques du piquet intermédiaire au piquet d'arrivée.

On peut représenter pour le cas où $n=7$ les tours de Hanoi aux trois étapes ci-dessus comme ceci :



Une astuce pour l'écriture du programme est de coder les piquets A, B, C par trois nombres 0, 1 et 2. On remarque que si A et C sont les piquets de départ et d'arrivée, $B = 3 - (A + C)$ nous donne le numéro du piquet intermédiaire.

16. Écrire le programme récursif qui résout le jeu des tours de Hanoi.
17. Calculer les complexités temporelle et spatiale de l'algorithme.

Il faut maintenant vérifier la correction de l'algorithme.

18. Justifier que la règle : "Un disque ne doit jamais être posé sur un disque de diamètre inférieur" est bien vérifiée par l'algorithme.
19. Modifier l'algorithme pour vérifier numériquement que c'est bien le cas à chaque étape.
20. Maintenant utilisez un ordinateur et vérifiez numériquement tous vos programmes et calculs.

Bonus : Réfléchir à ce qu'il se passe en terme d'écriture du programme et de complexité si la condition initiale diffère de "tous les disques sont sur le même piquet".