

TP : Les tours de Hanoï

Les tours de Hanoï sont un jeu de réflexion imaginé par le mathématicien français Édouard Lucas (1842-1891), et consistant à déplacer des disques de diamètres différents d'une tour de « départ » à une tour d'« arrivée » en passant par une tour « intermédiaire », et ceci en un minimum de coups, tout en respectant les règles suivantes :

- on ne peut déplacer plus d'un disque à la fois ;
- on ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide.

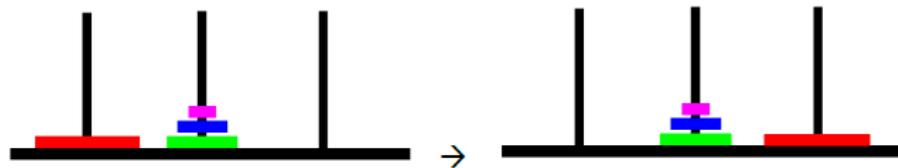


On suppose que cette dernière règle est également respectée dans la configuration de départ.

1. déplacement d'une pile de taille $n - 1$ de A vers B (C est la tour intermédiaire) :



2. déplacement d'un disque de A vers C (B est la tour intermédiaire) :



3. déplacement d'une pile de taille $n - 1$ de B vers C (A est la tour intermédiaire) :



Travail à faire

Utiliser le raisonnement décrit ci-dessus pour programmer une **procédure récursive** répondant à la spécification suivante :

```
def hanoi(n: int = 3, depart: list = [3, 2, 1], intermediaire: list = [],
arrivee: list = []):
    """Résout le problème des tours de Hanoï avec n disques
    et affiche les étapes successives"""
```

L'appel de cette procédure `hanoi(3, A, B, C)` avec les listes définies comme ci-dessus doit produire la sortie suivante :

```
[3, 2] -- [] -- [1]
[3] -- [2] -- [1]
[3] -- [2, 1] -- []
[] -- [2, 1] -- [3]
[1] -- [2] -- [3]
[1] -- [] -- [3, 2]
[] -- [] -- [3, 2, 1]
```

```

def hanoi(n,A,B,C,nb) :
    if n == 0 :
        print(dic)
        return nb + 1
    if n > 0 :
        nb = hanoi(n-1,A,C,B,nb)
        elt = A.pop(-1)
        C.append(elt)
        return hanoi(n-1,B,A,C,nb)

dic = {"A" : [3,2,1] , "B" : [] , "C" : []}
print(hanoi(len(dic["A"]),dic["A"],dic["B"],dic["C"],-1))

```

A l'exécution cela donne dans la console :

```

>>> (executing file "hanoi.py")
{'A': [3, 2, 1], 'B': [], 'C': []}
{'A': [3, 2], 'B': [], 'C': [1]}
{'A': [3], 'B': [2], 'C': [1]}
{'A': [3], 'B': [2, 1], 'C': []}
{'A': [], 'B': [2, 1], 'C': [3]}
{'A': [1], 'B': [2], 'C': [3]}
{'A': [1], 'B': [], 'C': [3, 2]}
{'A': [], 'B': [], 'C': [3, 2, 1]}
7

```

Si on réalise l'exécution :

```
dic = {"A" : [4,3,2,1] , "B" : [] , "C" : []}
print(hanoi(len(dic["A"]),dic["A"],dic["B"],dic["C"],-1))
```

On obtient :

```

>>> (executing file "hanoi.py")
{'A': [4, 3, 2, 1], 'B': [], 'C': []}
{'A': [4, 3, 2], 'B': [1], 'C': []}
{'A': [4, 3], 'B': [1], 'C': [2]}
{'A': [4, 3], 'B': [], 'C': [2, 1]}
{'A': [4], 'B': [3], 'C': [2, 1]}
{'A': [4, 1], 'B': [3], 'C': [2]}
{'A': [4, 1], 'B': [3, 2], 'C': []}
{'A': [4], 'B': [3, 2, 1], 'C': []}
{'A': [], 'B': [3, 2, 1], 'C': [4]}
{'A': [], 'B': [3, 2], 'C': [4, 1]}
{'A': [2], 'B': [3], 'C': [4, 1]}
{'A': [2, 1], 'B': [3], 'C': [4]}
{'A': [2, 1], 'B': [], 'C': [4, 3]}
{'A': [2], 'B': [1], 'C': [4, 3]}
{'A': [], 'B': [1], 'C': [4, 3, 2]}
{'A': [], 'B': [], 'C': [4, 3, 2, 1]}
15

```