

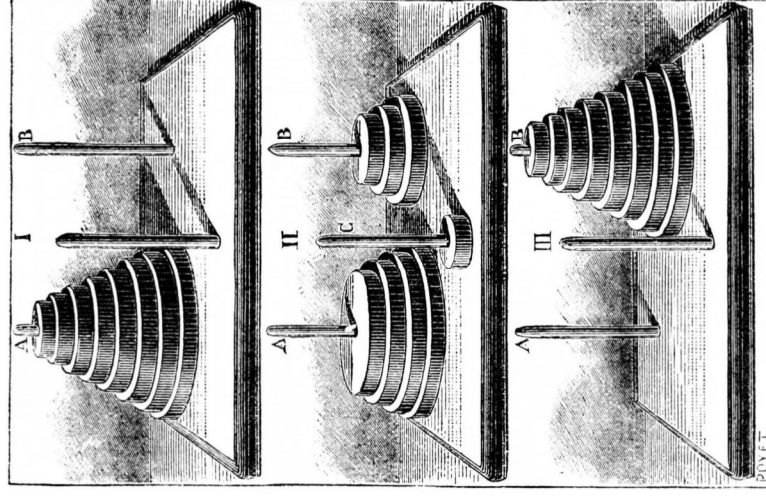
## 2.3. Tours de Hanoï



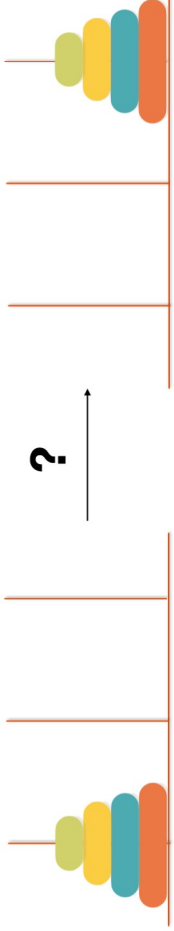
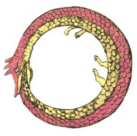
## Les tours de Hanoï

« ... dans le grand temple de Bénarès, au-dessous du dôme qui marque le centre du monde, trois aiguilles de diamant, plantées dans une dalle d'airain, hautes d'une coudée et grosses comme le corps d'une abeille.

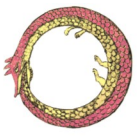
Sur une de ces aiguilles, Dieu enfila au commencement des siècles, 64 disques d'or pur, le plus large reposant sur l'airain, et les autres, de plus en plus étroits, superposés jusqu'au sommet. C'est la tour sacrée du Brahmâ. Nuit et jour, les prêtres se succèdent sur les marches de l'autel, occupés à transporter la tour de la première aiguille sur la troisième, sans s'écarter des règles fixes que nous venons d'indiquer, et qui ont été imposées par Brahma. Quand tout sera fini, la tour et les brahmes tomberont, et ce sera la fin des mondes! »



# Règles du jeu



- On déplace un seul disque à la fois.
- On ne peut pas placer un disque plus grand sur un plus petit



# Principe de la solution

**Cas trivial**  
**Pas de disque à déplacer**

Pour déplacer  $n$  disques de gauche à droite



Déplacer  $n-1$  disques de gauche au milieu



Déplacer le  $n^{\text{ème}}$  disque de gauche à droite

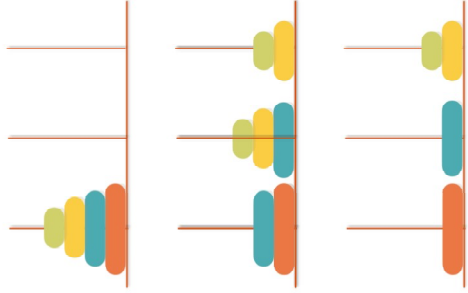
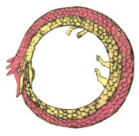


Déplacer  $n-1$  disques du milieu à droite

**Cas général**

**Cas général**

## Décomposons la première étape



Déplacer le disque de gauche au milieu

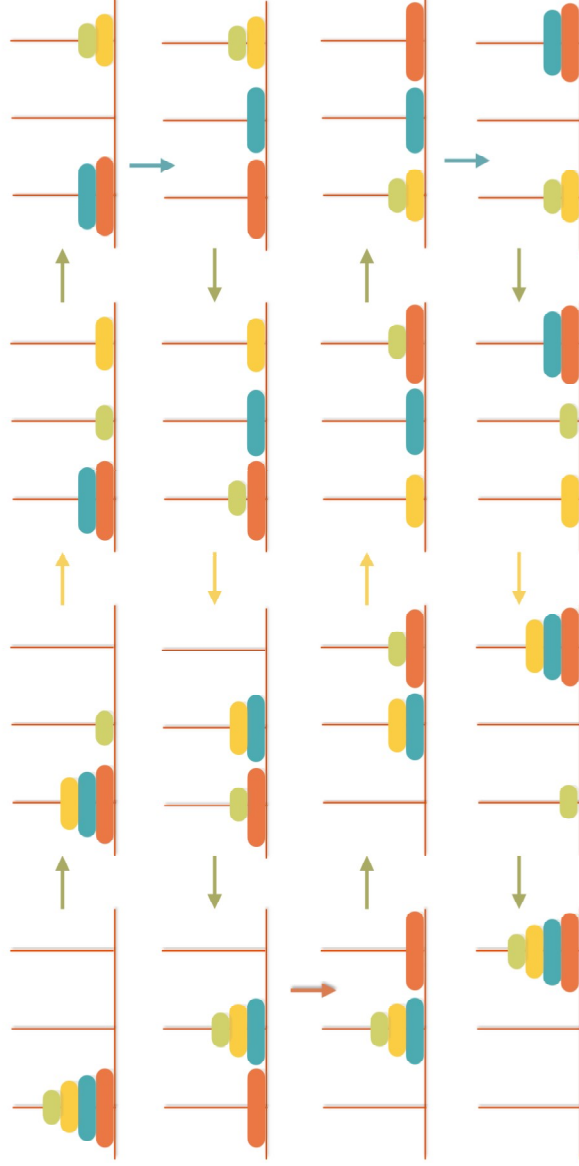


Déplacer le  $(n-1)^{\text{ème}}$  disque de gauche au milieu



Déplacer  $n-2$  disques de droite au milieu

## Toutes les étapes pour 4 disques



## CodeCheck it !

```
typedef vector<char> Tour;
vector<Tour> tours(3);

void transfert(Tour& from, Tour& via, Tour& to, int n)
{
    // A COMPLETER
    void transfert(Tour& from, Tour& via, Tour& to, int n)
    {
        if(n == 0) return;
        transfert(from,to,via,n-1);
        to.push_back(from.back());
        from.pop_back();
        display();
        transfert(via,from,to,n-1);
        // transfère n disques de la tour from à la tour to
        // en utilisant la tour via comme intermédiaire
        // appeler display() après chaque mouvement de disque.
    }
}
```

ASD - Algorithmes et Structures de Données - Printemps 2021

T0: A B C  
T1:  
T2:

T0: A B  
T1:  
T2: C

T0: A  
T1: B  
T2: C

T0: A  
T1: B C  
T2:

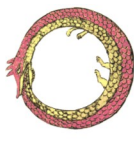
T0:  
T1: B C  
T2: A

T0: C  
T1: B  
T2: A

T0: C  
T1:  
T2: A B

T0:  
T1:  
T2: A B C

## Algorithme et complexité



### Algorithme :

Transférer n disques du piquet 0 (origine) vers le piquet D (destination) via le piquet I (intermédiaire):

**si** n > 0 **alors**

*Transférer* n-1 disques de 0 vers I via D

*transférer* le disque restant de 0 vers D

*Transférer* n-1 disques de I vers D via 0

**fin si**

### Complexité :

Soit  $T(n)$  le nombre de transferts pour n disques,

$$T(1) = 1$$

$$T(2) = 1 + T(1) + T(1)$$

$$T(n) = 1 + 2 T(n-1)$$

$$T(n) = 2^n - 1$$