

# Les tours de Hanoï

et la base trois

*Au XIX<sup>e</sup> siècle, Édouard Lucas a inventé le jeu des « tours de Hanoï », une simple récréation mathématique qui s'est révélée au fil des années une mine de réflexions. Nous nous intéressons ici à l'une d'elles : les liens avec les bases de numération.*

**Benoît Rittaud**  
Université Paris-13,  
Sorbonne-Paris-Cité



Le jeu des tours de Hanoï est constitué de trois piquets A, B et C, placés verticalement, et de  $n$  disques de taille décroissante. Chacun des disques est percé en son centre pour être mis autour de l'un ou l'autre des trois piquets. Les  $n$  disques sont initialement placés par taille décroissante autour du piquet A (celui de gauche), formant ainsi une tour. Le but du jeu consiste à déplacer les disques jusqu'à parvenir à la situation finale dans laquelle tous les disques se retrouvent autour du piquet C par ordre de taille décroissante. Les disques peuvent aller et venir librement sur les piquets, en suivant deux règles :

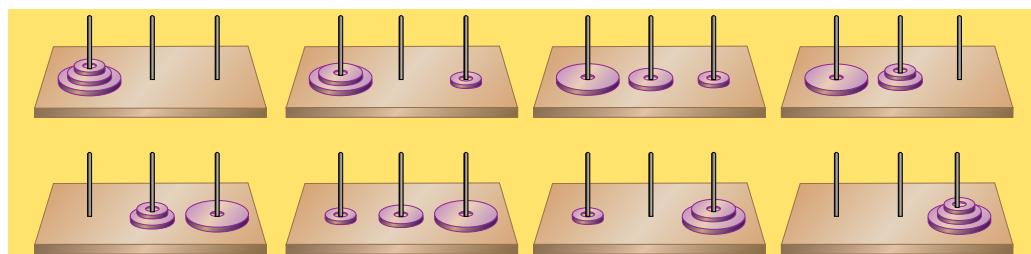
- on ne déplace qu'un seul disque à la fois ;
- un disque ne peut jamais être posé sur un disque plus petit.

En bas de page figure une solution dans le cas de trois disques.

Le problème n'est pas très difficile à résoudre dans le cas général (voir plus loin). Le contexte des tours de Hanoï n'en est pas moins d'une grande richesse. La preuve : mettez un combinatoriste devant et il trouvera aussitôt une question originale à se poser dessus. Tel a été le cas d'un collègue, Hacène Belbachir professeur à l'université d'Alger, qui m'a posé le problème suivant : « comment pourrait-on faire des tours de Hanoï en base trois ? »

## Une drôle de question

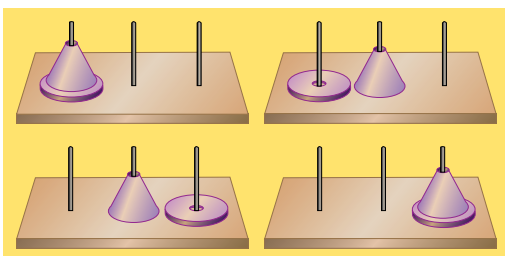
Vue de loin, la question est pour le moins bizarre. Quel est donc le rapport entre nos disques et une base de numération, quelle qu'elle soit ? Il faut connaître un petit peu le problème pour comprendre combien cette question est intéressante. Elle illustre comment une simple récréation mathématique peut donner l'occasion de réfléchir dans des domaines tout à fait inattendus.



En l'occurrence, les tours de Hanoï entretiennent un lien très fort avec la numération à base deux. Pour le comprendre, voyons tout d'abord un algorithme particulièrement bref pour résoudre le jeu. Cet algorithme est un cas d'école d'algorithme dit *récur­sif*, c'est-à-dire qui s'appelle lui-même au fil du parcours :

**Algorithme de résolution pour déplacer n disques d'un piquet d'origine à un piquet de destination**

- si  $n = 1$  : déplacer l'unique disque (situé sur le piquet d'origine) sur le piquet de destination, et c'est fini.
- si  $n > 1$  :
  - appliquer l'algorithme de résolution pour déplacer les  $(n-1)$  disques du piquet d'origine au piquet intermédiaire (qui n'est ni le piquet d'origine ni celui de destination finale, mais qui devient le nouveau piquet de destination dans cette étape).
  - déplacer le  $n$ -ième disque sur le piquet de destination ;
  - appliquer à nouveau l'algorithme de résolution pour déplacer les  $(n-1)$  disques du piquet intermédiaire au piquet destination. Le piquet intermédiaire devient le nouveau piquet origine pour cette étape.



L'essentiel ici est que cet algorithme est optimal (c'est-à-dire le plus rapide possible pour résoudre le jeu), en raison de l'impératif suivant : il faut à un moment ou à un autre faire la place pour pouvoir déplacer le gros disque du dessous, ce qui impose d'avoir déplacé préalablement les  $n-1$  plus petits sur un seul et même piquet – c'est-à-dire, donc, d'avoir résolu préalablement le problème des

## Numéroter dans d'autres bases

Notre façon usuelle d'écrire les nombres est dite « à base dix », car nous utilisons dix chiffres (de 0 à 9) pour écrire tous les nombres. En base dix, une expression telle que 548 correspond à la valeur « huit unités plus quatre dizaines plus cinq dizaines de dizaines ». Les premiers entiers en base dix sont inscrits dans la première colonne du tableau.

Le choix d'utiliser dix chiffres plutôt que sept, treize ou soixante est issu d'habitudes anciennes (qui découlent sans doute du fait que nous avons dix doigts) et non d'un impératif mathématique. Or certaines situations rendent plus utiles le recours à d'autres bases. L'informatique, par exemple, est une grosse consommatrice de la base deux, dans laquelle seuls deux chiffres sont utilisés : 0 et 1. La deuxième colonne du tableau donne l'écriture en base deux des premiers nombres naturels.

Dix	BASE	
	Deux	Trois
0	0	0
1	1	1
2	10	2
3	11	10
4	100	11
5	101	12
6	110	20
7	111	21
8	1000	22
9	1001	100
10	1010	101
11	1011	102
12	1100	110
13	1101	111
14	1110	112
15	1111	120
16	1 0000	121
...	...	...

Ainsi pour une même ligne, les valeurs dans les différentes colonnes renvoient aux écritures dans différentes bases d'un même nombre.

Le principe qui permet de passer d'un entier au suivant en base deux est le même qu'en base dix, la seule différence étant le moment des retenues : en base dix, il y a une retenue dès qu'on dépasse neuf, en base deux il y en a une dès qu'on dépasse un.

La valeur correspondant à une expression en base deux s'obtient selon le même principe qu'en base dix, les paires remplaçant les dizaines, les « quatraines » (i.e. les paires de paires) remplaçant les centaines (i.e. les dizaines de dizaines), et ainsi de suite. Par exemple, l'expression 1001011 en base deux correspond à la valeur que nous calculerions en base dix en écrivant

$$1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

(qui vaut soixante-quinze et s'écrit, donc, 75 en base dix).

En base trois, le principe est le même. Les nombres s'écrivent cette fois avec les chiffres 0, 1 et 2. La liste des premiers entiers en base trois est donnée dans la troisième colonne du tableau ci-haut à droite.

Une expression en base trois telle que 10212 correspond à ce que nous écririons, dans la bonne vieille base dix,

$$1 \times 3^4 + 0 \times 3^3 + 2 \times 3^2 + 1 \times 3^1 + 2 \times 3^0$$

(qui vaut cent quatre et s'écrit donc 104 en base dix, si vous voulez le savoir...).

Plus généralement, à tout entier  $b > 1$  fixé correspond un système de numération à base  $b$ .

## La récursivité

Un algorithme est une méthode systématique de résolution d'un problème donné. Il peut être *itératif* ou *récursif*. Les algorithmes itératifs sont ceux auxquels on est le plus souvent habitués, ils consistent en une suite d'instructions faisant appel à des objets définis par ailleurs. Une recette de cuisine est un exemple d'algorithme itératif.

Un algorithme récursif, quant à lui, a la propriété de s'appeler lui-même au fil de son déroulement. Un grand intérêt des algorithmes récursifs est qu'ils sont en général plus courts et synthétiques que les algorithmes itératifs. Leur inconvénient est l'encombrement de mémoire qu'ils peuvent engendrer.

Un cas d'école pour comparer les deux types d'algorithmes est le calcul de la factorielle d'un entier. Rappelons que, pour tout entier positif  $n$ , l'expression  $n!$  (« factorielle  $n$  ») désigne le produit des entiers de 1 à  $n$ . On a ainsi, par exemple  $5! = 120$  ( $= 1 \times 2 \times 3 \times 4 \times 5$ ), ou  $7! = 5040$  ( $= 1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7$ ), etc. Par convention, on pose aussi que  $0! = 1$ .

Un algorithme itératif de calcul de  $n!$  est le suivant :

### Factorielle itératif ( $n$ )

```
p := 1
pour i de 1 à n :
    p := p × i
retourner(p)
```

Et voici un algorithme récursif, qui utilise le fait que, pour calculer  $n!$ , il suffit de multiplier par  $n$  la valeur  $(n-1)!$  :

### Factorielle Récursif ( $n$ )

```
si n = 0 :
    retourner(1)
sinon :
    retourner(n × FactorielleRécursif(n-1))
```

Lorsque le problème est suffisamment simple, il est facile d'en programmer la résolution aussi bien de manière itérative que récursive. Parfois en revanche, le point de vue récursif est beaucoup plus simple. Le cas des tours de Hanoï illustre bien les avantages et inconvénients de la récursivité : d'un côté l'algorithme récursif de résolution est extrêmement bref et clair, d'un autre côté il n'est pas très maniable pour un humain qui est, lui, plus volontiers preneur d'un algorithme itératif pour ne pas se perdre dans la gestion de la pile des tâches qu'il lance.

tours de Hanoï pour ces  $n-1$  disques. Cette remarque permet facilement de se persuader que l'algorithme précédent est bien optimal.

### Combien d'étapes ?

En prime, notre algorithme nous fournit un moyen explicite pour compter le nombre total de mouvements. Si  $u_n$  désigne ce nombre pour le problème à  $n$  disques, alors les trois

étapes de l'algorithme (résolution pour  $n-1$  disques, puis déplacement du gros disque, puis à nouveau résolution pour  $n-1$  disques) conduisent à la formule de récurrence :

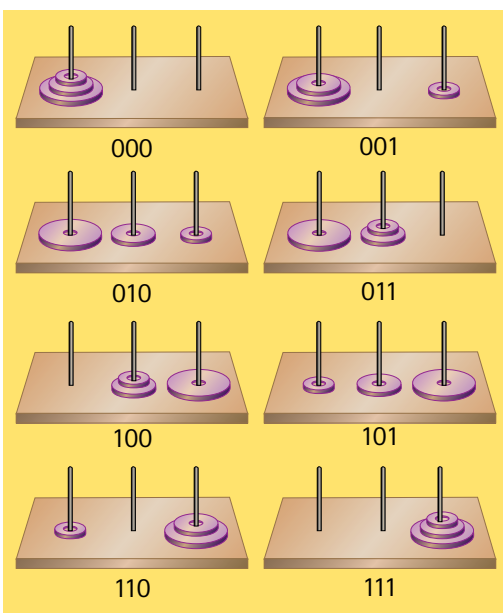
$$u_n = u_{n-1} + 1 + u_{n-1}$$

c'est-à-dire  $u_n = 2u_{n-1} + 1$ . L'initialisation de l'algorithme donnant par ailleurs que  $u_1 = 1$ , les premiers termes de la suite sont donc :

$$1, 3, 7, 15, 31, 63, 127...$$

On reconnaît ici, à une unité près, la suite des puissances de deux, ce qui se démontre sans difficulté par récurrence.

Le nombre d'étapes étant donc de  $u_n = 2^n - 1$ , le nombre d'états (entre l'état initial et l'état final) est lui de  $2^n$  : voilà poindre, enfin, notre lien avec la base deux. En effet, numérotons les états en base deux (l'état initial portant le numéro zéro). Ce faisant, on réalise une correspondance entre les états successifs et l'ensemble des entiers qui peuvent s'écrire avec exactement  $n$  chiffres en base deux. Rien de si extraordinaire jusque là, sauf que... voici ce que cela donne par exemple pour trois disques. Une belle surprise nous attend.



La surprise, c'est que le numéro de chaque état porte une information capitale : à l'état  $i$ , la position du chiffre le plus à gauche qui a changé en passant de  $i-1$  à  $i$  donne le numéro du disque qui vient d'être déplacé. Par exemple, l'état 110 fait suite à l'état 101 : l'addition d'une unité dans le passage de 101 à 110 a, par le mécanisme des retenues, modifié jusqu'au deuxième chiffre (le 0 de 101 est devenu un 1 dans 110). Et c'est en effet le deuxième disque qui a été déplacé lors du passage de l'état 101 à l'état 110. Ce phénomène spectaculaire fonctionne quel que soit le nombre de disques. (Exercice : le démontrer !)

### De deux à trois

Le lien entre les tours de Hanoï et la base deux est donc tout sauf superficiel. Entre autres choses, il permet de produire un algorithme itératif de résolution, alternatif à l'algorithme récursif présenté plus haut. (Exercice : l'écrire.) Il permet aussi de montrer diverses propriétés que vous pourrez facilement vérifier, comme par exemple le fait que le  $i$ -ième disque est, en tout, déplacé exactement  $2^{n-i}$  fois au fil du jeu.

À présent, donc, nous voilà prêts à comprendre la question à l'origine de cet article : comment modifier les règles du jeu pour que sa résolution fasse « naturellement » apparaître la base trois plutôt que la base deux.

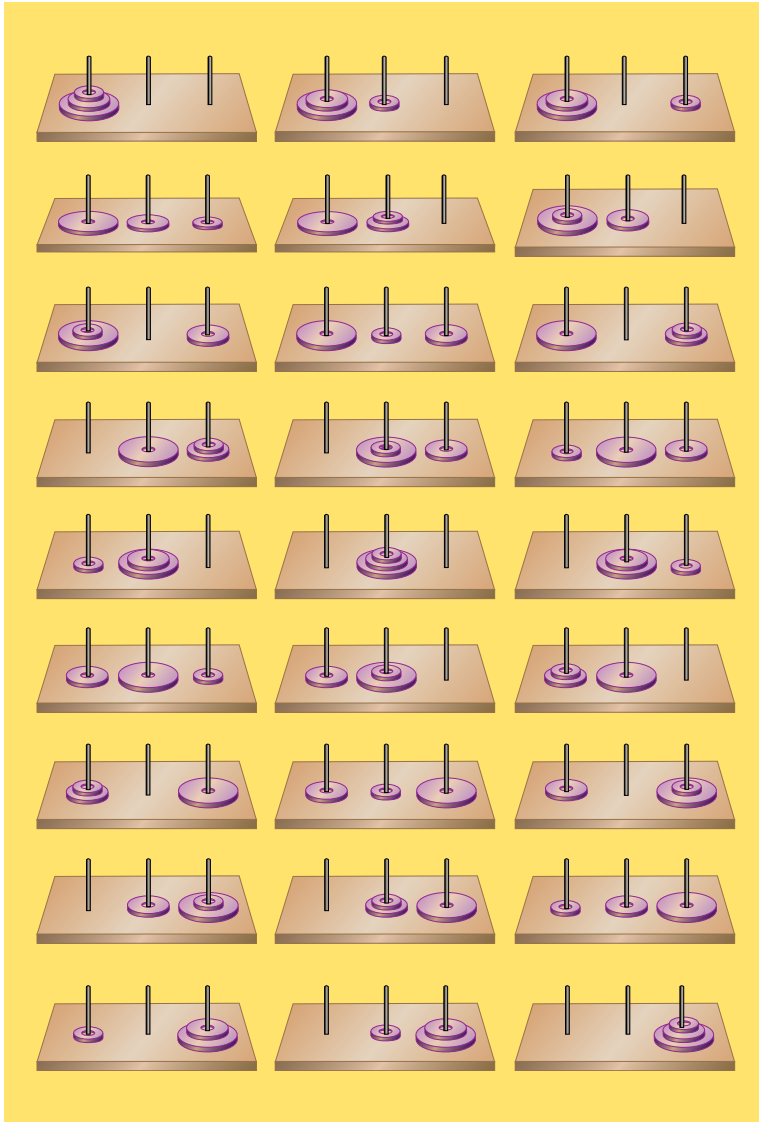
Pour ce qui est de modifier les règles, on n'a que l'embaras du choix : augmenter le nombre de piquets, poser des règles exotiques d'empilement des disques, interdire certains mouvements... Dans le domaine, beaucoup de questions sont d'ailleurs encore ouvertes.

Une façon qui me semble très satisfaisante pour faire apparaître la base trois est celle-ci :

*On interdit aux disques d'aller directement de A à C ou de C à A.*

(On suppose que tous les disques sont au départ sur A et que l'objectif est de les déplacer sur C.) En d'autres termes, à chaque fois qu'on voudrait faire passer un disque de A à C, on s'oblige à le faire passer d'abord par B.

Voici, à la page suivante, la résolution de cette variante dans le cas de trois disques :



Il faut 26 étapes, c'est-à-dire  $3^3-1$ . Si vous avez bien suivi les explications du cas classique, alors vous vous doutez probablement de ce qui va suivre. Commençons par l'algorithme récursif (optimal) de résolution de ces « tours de Hanoï à  $n$  disques en base trois » :

*Algorithme de résolution pour déplacer  $n$  disques d'un piquet origine à un piquet sans sauter par-dessus un autre piquet.*

- Si  $n = 1$  : Si les piquets origine et destination sont voisins : déplacer ce disque unique de l'origine à la destination ;  
SINON (i.e. s'il y a un piquet entre les deux) déplacer ce disque unique de l'origine au piquet intermédiaire ;  
Appliquer l'algorithme pour déplacer ce disque unique du piquet intermédiaire vers le piquet destination
- Si  $n > 1$  :
  - Appliquer l'algorithme pour déplacer les  $(n-1)$  disques du dessus, du piquet-origine au piquet-destination sans sauter par-dessus un piquet
  - déplacer le  $n$ -ième disque au piquet intermédiaire ;
  - Appliquer l'algorithme pour déplacer les  $(n-1)$  disques du dessus, du piquet-destination au piquet-origine sans sauter de piquet ;
  - déplacer le  $n$ -ième disque au piquet destination ;
  - Appliquer l'algorithme pour déplacer les  $(n-1)$  disques du dessus, du piquet-origine au piquet-destination sans sauter par-dessus un piquet.

Le nombre  $v_n$  d'étapes nécessaires satisfait cette fois la formule de récurrence :

$$v_n = v_{n-1} + 1 + v_{n-1} + 1 + v_{n-1},$$

soit  $v_n = 3v_{n-1} + 2$ . L'initialisation de l'algorithme donnant  $v_1 = 2$ , on vérifie par une simple récurrence que  $v_n = 3^n - 1$  pour tout  $n$ . La suite est alors toute tracée :  $3^n - 1$  étapes font  $3^n$  états. La numérotation des étapes en base trois dans le cas de trois disques est donnée dans la figure de la page de droite.

Le « miracle » qui s'était produit dans le cas classique avec la base deux opère aussi pour notre variante, mais cette fois avec la base trois : on lit sur le numéro de l'état  $i$  le disque qui a été déplacé (c'est celui qui correspond à la place du chiffre le plus à gauche qui a été modifié lors du passage de l'état  $i-1$  à l'état  $i$ ).

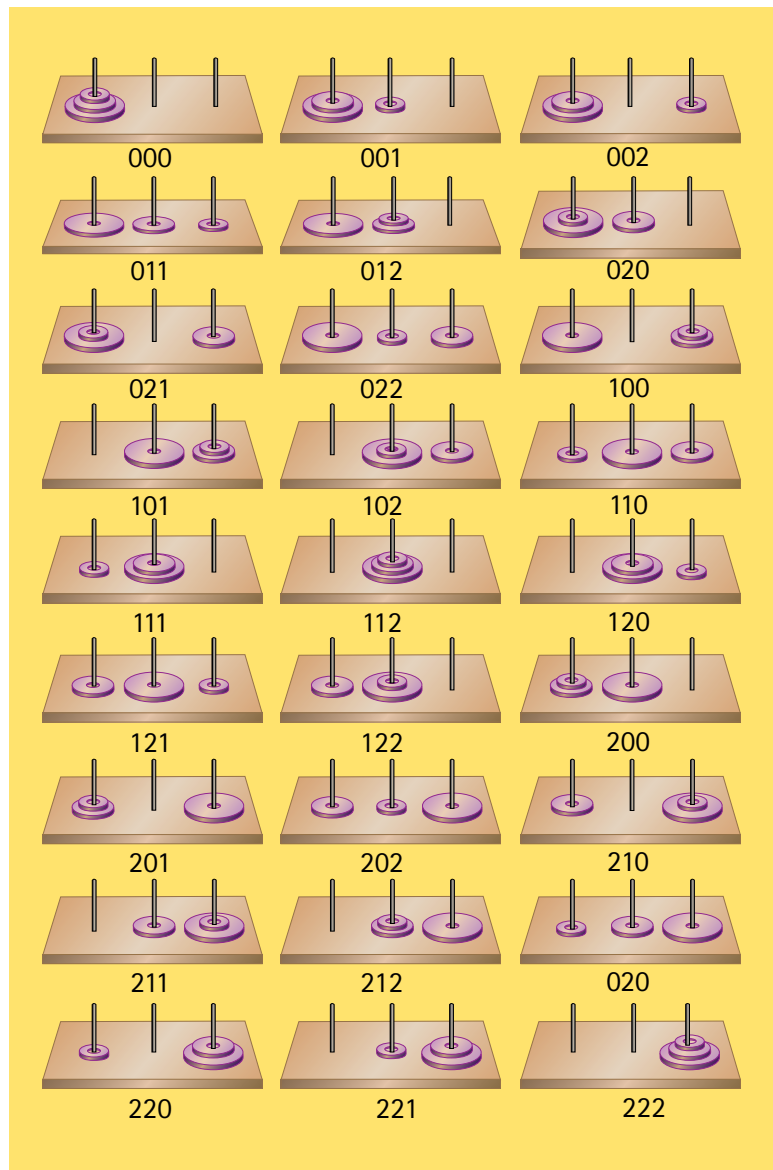
Un peu de travail supplémentaire vous indiquera aussi, tout comme dans le cas classique, les piquets concernés par le déplacement, ce qui permet là aussi de produire un algorithme itératif.

Cette variante des tours de Hanoï présente un intérêt particulier : elle permet de parcourir la totalité des états admissibles, c'est-à-dire qui satisfont à la règle qu'un disque plus léger ne porte jamais un disque plus lourd. En effet, le nombre de ces états admissibles est égal au nombre de façons de constituer un mot de  $n$  lettres de l'alphabet  $\{A, B, C\}$  (la  $i$ -ième lettre du mot indiquant le piquet sur lequel se trouve le disque  $i$ ), qui vaut précisément  $3^n$ .

### Autres questions

Puisque la variante à base trois « consomme » tous les états possibles, s'interroger sur des tours de Hanoï à base quatre (ou plus) obligerait à innover encore dans les règles. J'avoue ne pas savoir comment. D'autres questions sont également possibles, plus ou moins difficiles, et sur lesquelles le lecteur pourra méditer :

- En combien de coups peut-on changer une tour de  $n$  disques initialement en A en deux tours, l'une en B contenant tous les disques pairs et l'autre en C contenant tous les disques impairs ?
- À partir de l'écriture en base deux (à  $n$  chiffres) d'un nombre  $a$ , comment reconstituer la position des disques à l'état du jeu correspondant ? (Même question, bien sûr, pour la variante à base trois...)
- À partir de l'écriture en base trois d'un nombre  $a$ , est-il possible de savoir si la configuration correspondante des disques apparaît ou non dans la résolution du jeu classique ?
- Étudier d'autres variantes, telles que la suivante : le déplacement direct de A à C est interdit, mais tous les autres autorisés – y compris, donc, celui de C à A. (Attention, il y a là du calcul matriciel en dimension 6...)



1. Dans ce raisonnement, les piquets A, B et C sont supposés non équivalents, alors que dans les présentations du cas courant, les piquets B et C sont parfois considérés comme interchangeables. Dans cette dernière hypothèse, le nombre de configurations non-équivalentes est divisé par deux (chaque configuration allant de paire avec celle dans laquelle les disques des piquets B et C sont échangés), ce qui fait que, même avec ces équivalences, la résolution du jeu classique en  $2^n - 1$  étapes ne fait pas décrire tous les états possibles.