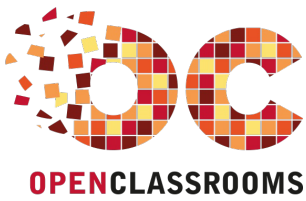
Les tours de Hanoï

Par Olivier Strebler (shareman)



www.openclassrooms.com

Dernière mise à jour le 2/06/2009

Sommaire

Les tours de Hanoï

Par  Olivier Strebler (shareman)

Mise à jour : 02/06/2009
Difficulté : Intermédiaire

Le problème des tours de Hanoï est un jeu faisant parti de la catégorie des casse-tête.
Il est très utilisé en psychiatrie, étudié en mathématiques et en algorithmique et souvent utilisé pour montrer la puissance et l'intérêt de la récursivité.
Sommaire du tutoriel :

Formalisation

Le problème des tours de Hanoï est un problème mathématique célèbre inventé par le mathématicien français Édouard Lucas en 1883 qu'il devrait à un de ses amis, N. Claus de Siam. C'est également un jeu de réflexion intéressant. Le mathématicien écrit :

"*N. Claus de Siam a vu, dans ses voyages pour la publication des écrits de l'illustre Fer-Fer-Tam-Tam, 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 !*"

Le problème, qui peut devenir très vite complexe, peut être décrit très simplement. On dispose de trois "tours" A, B et C formées d'un empilement plus ou moins grand de disques de telle sorte que chaque disque ait un diamètre inférieur à son prédécesseur. Ainsi, le plus grand disque de chaque tour se situe à leur base et le plus petit sur leur sommet. Au départ, tous les disques (supposons qu'il y en ait n) se trouvent empilés suivant les règles précédemment décrites sur la tour A. L'objectif est de déplacer tous les disques de la tour A vers la tour C en s'aidant uniquement de la tour B, tout en respectant les règles suivantes :

- On ne peut bouger qu'un seul disque par étape et cela doit toujours être le plus petit (celui qui se trouve au sommet) ;
- À chaque étape, la règle d'organisation des tours décrite précédemment doit être respectée.

Solution récursive

La résolution du problème des tours de Hanoï est très étudiée en algorithmique où elle sert notamment à montrer que l'utilisation de la récursivité pour résoudre de gros problèmes peut produire des codes à la fois logiques, puissants et concis. L'intérêt de la récursivité sera donc particulièrement mis en avant dans cette sous-partie. Si vous n'avez pas la moindre idée de ce qu'il en retourne quand je dis "récursivité" ou si vous avez des doutes sur les performances des implémentations récursives, je vous recommande ce tutoriel assez complet.

Pour résoudre le problème des tours de Hanoï, il faut procéder logiquement. Nous allons partir d'un exemple trivial, presque atomique :

Nous disposons de trois tours A, B et C. Deux disques sont sur A, les tours de départ et B et C sont vides. Comment s'y prendre pour déplacer tous les disques de A vers C en passant par B ?

Les règles d'organisation des tours de Hanoï, à savoir que chaque disque possède un diamètre inférieur au disque situé en-dessous de lui, doivent toujours être respectées. Nous savons que sur la tour A, le plus grand disque se situe sous le second, qui est forcément plus petit. Or, nous devons parvenir à un résultat tel que ce plus grand disque se trouve à la base de la tour C et le second, plus petit, au sommet de la tour C ; et malheureusement, nous sommes contraints de ne déplacer qu'un disque à la fois et toujours de sommet vers sommet. La première opération à faire est donc de dégager le plus petit disque (le sommet de la tour A) et de l'empaler dans la tour B. On peut ensuite aisément déplacer la base de la tour A, le plus grand disque, vers la base de la tour C. Puis il suffit de retirer le petit disque de la tour B et de le placer au sommet de la tour C. Au final, on a déplacé tous les disques de la tour A vers la tour C, le problème a été résolu correctement.

L'affaire se complique un peu quand on se retrouve avec plus de deux disques. Si j'ai choisi de commencer par là, c'est uniquement parce que la résolution des tours de Hanoï avec deux disques permet de trouver la logique qui va nous permettre de résoudre des tours de Hanoï avec un nombre supérieur de disques.

Il faut penser récursivement. Pour résoudre le problème des tours de Hanoï avec deux disques, on a déplacé de A vers B, puis de A vers C pour finalement déplacer de B vers C (une seule et unique opération est à chaque fois possible). Donc (2-1) disque de A vers B, puis 1 disques (le plus grand) de A vers C et à la fin (2-1) disque de B vers C. La logique est donc la suivante : "isoler le plus grand disque de A puis le placer à la base de la tour de destination C puis appliquer le même algorithme de B vers C". En résumé, l'algorithme de résolution des tours de Hanoï sur un nombre n de disques est le suivant :

- Déplacer n disques de A vers C en passant par B :
  - Déplacer (n-1) disques de A vers B en passant par C;
  - Déplacer 1 disque de A vers C ;
  - Déplacer (n-1) disques de B vers C en passant par A.

Nous pouvons aisément implémenter cet algorithme dans n'importe quel langage de programmation, tant que ce dernier permet la récursivité. Je vous propose ici une implémentation, toute simple, en OCaml. La fonction (`let `) récursive (`rec `) hanoi prend en paramètre n, le nombre de disque(s) à déplacer, a qui symbolise la tour de départ A, c représentant la tour de destination C et b qui est la tour intermédiaire B. Cette fonction est uniquement là à titre pédagogique et n'a pas de réel intérêt si ce n'est le calcul exact du nombre de déplacements effectués (1 représente un déplacement).

Code : OCaml

```
let rec hanoi n a c b = match n with
| 1 -> 1 (* Déplacer 1 disque de a vers c *)
| _ -> hanoi (n-1) a b c + hanoi 1 a c b + hanoi (n-1) b c a
```

Il peut également être intéressant d'utiliser les big_int d'OCaml. En effet, comme nous allons le voir, le nombre de déplacements nécessaires peut vite devenir très grand.

Code : OCaml

```
let rec hanoi n a c b = match n with
| 1 -> Big_int.unit_big_int (* Déplacer 1 disque de a vers c *)
| _ -> Big_int.add_big_int
        (Big_int.add_big_int (hanoi (n-1) a b c) (hanoi 1 a c b))
        (hanoi (n-1) b c a)
```

L'évaluation de la complexité de cet algorithme est assez simple. Nous allons chercher à savoir combien de déplacements de disques sont nécessaires pour arriver à nos fins. Très simplement, la fonction calculant ceci peut être définie de la manière suivante (on peut se baser sur le code OCaml ci-dessus) : f(1) = 1; f(n) = 2*f(n-1)+1. On peut démontrer par récurrence que ceci est égal à f(n) = 2^n-1 : On a f(n+1) = 2*f(n)+1. En supposant que f(n) = 2^n-1, on a f(n+1) = 2*(2^n-1)+1. On développe ensuite en f(n+1) = 2*2^n-2+1 ce qui nous donne au final f(n+1) = 2^(n+1)-1. On a ainsi démontré que si c'est vrai pour n, alors c'est vrai pour (n+1). Comme on a f(1) = 2*0+1 = 2^1-1 = 1, c'est vrai pour tous les n supérieurs à 1 et 1. La complexité exacte de l'algorithme présenté ci-dessus est donc en 2^n-1 pour n disques à déplacer de la tour A vers la tour C. Avec la notation de Landau, on dit qu'il s'agit là d'une complexité exponentielle, c'est à dire en O(2^n).

Aller plus loin

Nous venons de voir ce qu'est réellement de problème des tours de Hanoï, nous avons étudié ensemble une solution récursive à ce problème que nous avons ensuite évaluée en calculant sa complexité. Mais ce n'est pas encore fini ! Il y a encore tant à voir, d'autres solutions à étudier et à évaluer, etc. Je vous propose donc ici quelques liens qui vous permettrons l'aller plus loin.

- [Wikipédia] Solution itérative ;
- [Wikipédia] Représentation à l'aide d'un graphe ;
- [Wikipédia] Article wikipédia, très complet, en anglais ;
- [/] Article assez complet, avec animation ;
- [Prise 2 Tête] Animation, à résoudre soi-même ;
- [Mazeworks] Applet Java, à résoudre soi-même ;
- [Cut the Knot] Animation et article assez complet ;
- [Wikipédia] Utilité ?

Le but de ce tutoriel, relativement court, est de vous présenter brièvement le problème des tours de Hanoï et une méthode de résolution, la récursive. La dernière sous-partie de ce cours est donc également très importante : elle vous fait par exemple découvrir des solutions comme la solution itérative, beaucoup moins évidente.

En espérant que ce cours vous a intéressé et vous en a appris, je vous souhaite une bonne continuation !

Je remercie Nanoc pour ses explications sur la démonstration par récurrence ainsi que zAmnellL pour sa relecture attentive.

ShareMan