# Hanoi Towers

This excellent game was created by French mathematician Edouard Lucas in 1883 under the name of N. Claus de Siam. Legend says there is a temple in which Hindu monks constantly move 64 golden disks over three columns made of diamond, following the rules of Tower of Hanoi (also known as Tower of Brahma) and that the world will end when the monks will finish moving the disks.

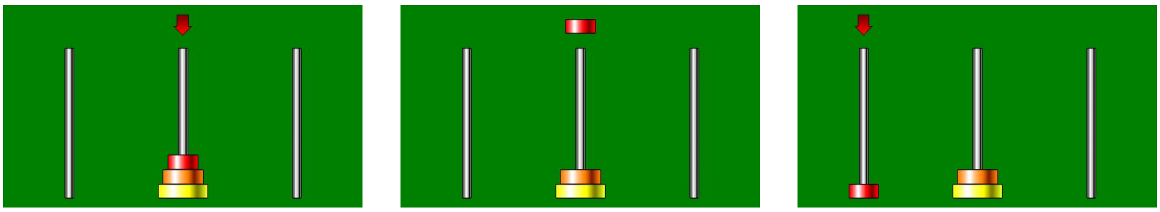
**Some real Tower of Hanoi sets**

The most curious thing is that by increasing the number of disks you have to repeat simpler operations already seen for smaller disks (and mathematically there are many ways to show this..), but this time the aim is to build a taller tower so all this moving from a place to another will soon become less intuitive and recursive than it would ever seem at first glance!!

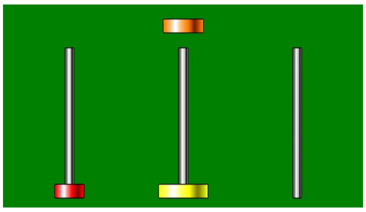
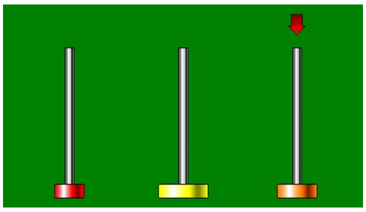
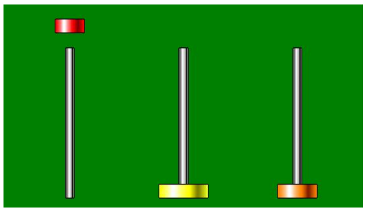
Basically a human player will develop many personal tricks to understand the logic of this game and despite the variety of these tricks the way of solving the entire tower in the lesser number of moves will look quite identical.

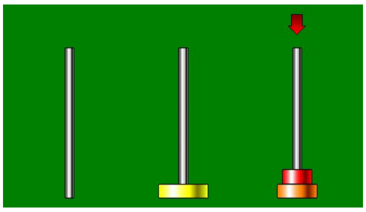
Case with disks = 1, 2 is banal; anyway with two disks it's yet possible to understand what I called

**rule 2: "If you have to move an even pile to X, start moving the first disk to Y; if the pile is odd, move it to X"**



*by Polarsun*



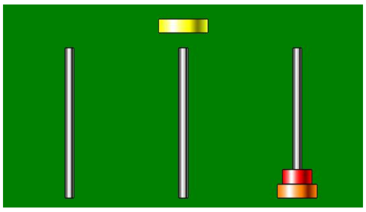
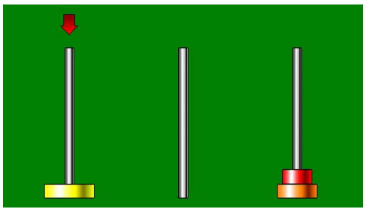
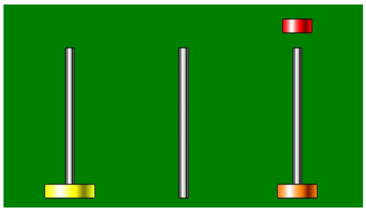
The reason of this is that if you have more than one disk to place, you will have two main aims to remember during the whole game: to put the “base” of the new tower (orange disk) in the correct column and then to rebuild the full tower in the choosen column; since you can’t put a bigger disk over a smaller one, this is what really happens: a pile made of 1 disk jumps to a column, then a pile of 2 disks is reconstructed over the other column and so on with piles made of 3, 4, 5 and more disks; if your first move is wrong, the whole pile will go to the wrong column..

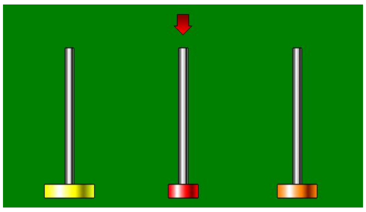
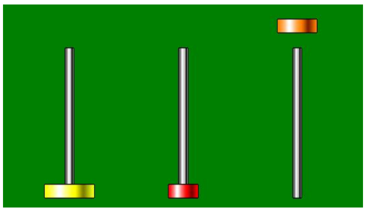
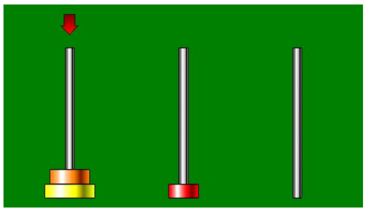
The confusion comes up when you don’t understand anymore what is being destroyed in order to be reconstructed in another place! so there is

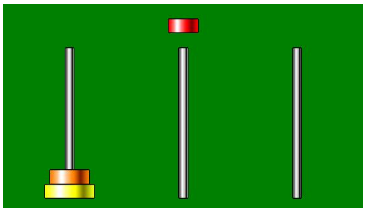
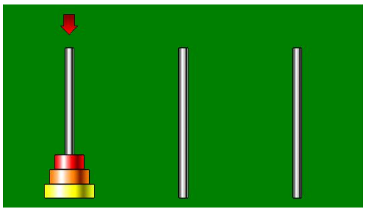
**rule 1: “There is always a main objective in what you are doing”**

That is a joke only for the computer, because it remembers all the encapsulated objectives and the structure of moves to do! Human player instead wants some fun so he will do a good bunch of mistakes..

So for example with d = 2 the main objective was to build a pile of 2 disks in column 3 (as shown in figures) and also the intermediate objective of putting the orange base on column three was necessary. Now let’s imagine d = 3 with the same figures.. following **rule 2** what we did is also correct for moving a pile of 3 disks from 2nd to 1st column: we initially wanted to move a pile of 2 disks to column 3, so because of the odd tower we moved a pile of 1 disk to the first column, then we were forced to move the orange disk to the third column and the red over the orange to “free” the yellow disk. Now we will move the yellow to first column in order to put the new “base”. We could call this objective “free the base”! Completing the tower has become banal, because it consists in moving a pile of 2 from third column to first one.

These apparently simple rules are all what must be known to go further with more disks.. the game gets harder but it becomes clear that everything is going fine if you free the base, then you manage to put another piece over it (the new base), then another base.. till reaching the top and doing always the same things in a very intricated way.. you reach the point of swapping logic and creativity with this game.

Let's see another interesting point of view over the game: which kind of complete towers are built while building a bigger one? Solving the game, this is the answer:

| ***disks*** | ***heights of full piles (sequence)*** |
|---|---|
| *1* | 1 |
| *2* | 1 2 |
| *3* | 1 2 1 3 |
| *4* | 1 2 1 3 1 2 1 4 |
| *5* | 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1 5 |

So apparently for a (n)-disks tower we do the same moves that are for a (n-1)-disks, then of (n-2)-disks, etc. moving continuosly the smallest disk and disassembling all the structure till the elemental piece; this could also seem quite illusory and contradictory, because at the same time of disassembling something, we are putting bases and a whole new tower is rebuilt!

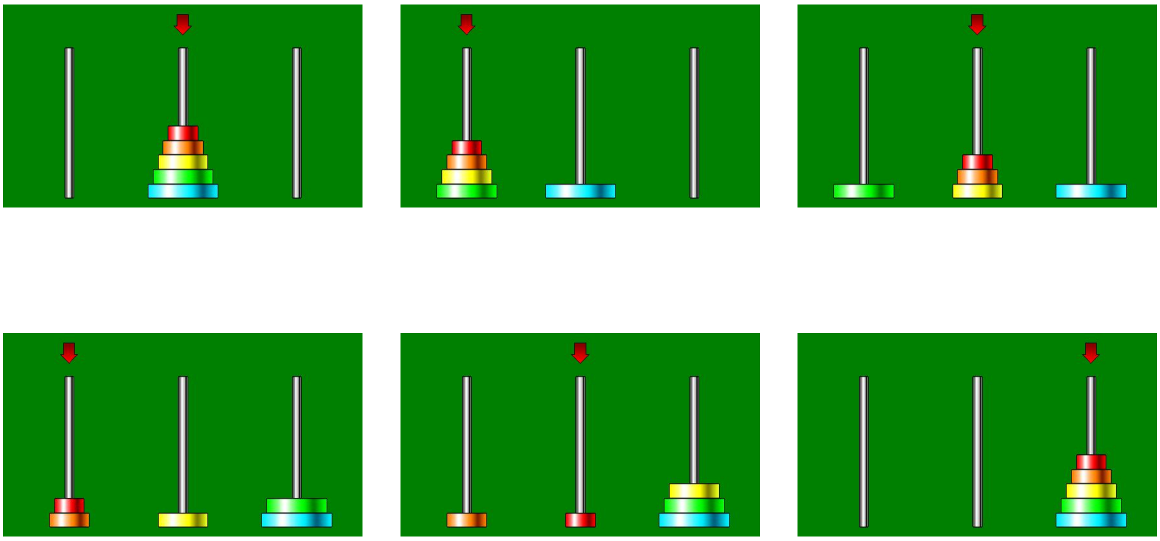
*by Polarsun*

This sort of excursus can explain what must be done to completely move a tower made of many disks; surely the first part of the 5-disks tower sequence above tells you that a full pile of 4-disks will be created somewhere; so it is good to simplify this thing saying something very similar to what was seen in 3-disks exercise:

**rule 3:**

**“To move a (n)-disks stack with n>3:**

- **move a (n-1) pile, then free the base**
- **move a (n-2) pile, then free another base and put it on the previous base**
- **..till you put the last disk, the smallest one, over the top of the tower!”**



*An iterative approach*

## P3

| | | | |
|---|---|---|---|
| *1-2* | *P3 to 2* \| | *P2 to 2* | |
| | | *P1 to 1* | *free disk 3* |
| | *P2 to 3* \| | *P1 to 3* | *free disk 2* |
| | *P1 to 2* | | |
| *1-3* | *P3 to 3* \| | *P2 to 3* | |
| | | *P1 to 1* | *free disk 3* |
| | *P2 to 2* \| | *P1 to 2* | *free disk 2* |
| | *P1 to 3* | | |
| *2-1* | *P3 to 1* \| | *P2 to 1* | |
| | | *P1 to 2* | *free disk 3* |
| | *P2 to 3* \| | *P1 to 3* | *free disk 2* |
| | *P1 to 1* | | |
| *2-3* | *P3 to 3* \| | *P2 to 3* | |
| | | *P1 to 2* | *free disk 3* |
| | *P2 to 1* \| | *P1 to 1* | *free disk 2* |
| | *P1 to 3* | | |
| *3-1* | *P3 to 1* \| | *P2 to 1* | |
| | | *P1 to 3* | *free disk 3* |
| | *P2 to 2* \| | *P1 to 2* | *free disk 2* |
| | *P1 to 1* | | |
| *3-2* | *P3 to 2* \| | *P2 to 2* | |
| | | *P1 to 3* | *free disk 3* |
| | *P2 to 1* \| | *P1 to 1* | *free disk 2* |
| | *P1 to 2* | | |

### Explanation

reading is in reverse order from last line to first line and from right to left

"P3" means "a Pile of 3 disks"

"x-y" means "from column x to y"

"P2 to 3" means "sub-objective is to build a full Pile of 2 disks in column 3 by moving only disk 1 and disk 2"

### Considerations

coherence of approach can be deducted visually,

ex. P3 (1-2), P3 to 2 block:

- to move a Pile of 3 disks from column 1 to column 2 the final objective is "P3 to 2"
- "P2 in 2" and "P3 in 2" conditions are achieved at the same time
- P1 is forced to go to 1 because of lower and higher conditions: it is in 3rd column (P2 to 3) and must contribute to an even pile on column 2 (P2 to 2); **rule 2** says it must go anywhere from where it is (3) but not in (2), so in (1)

ex. P3 (1-2), main objectives:

- m.o. also follow **rule 2**: from 1, P1 goes to 2 (final destination), alternatively P2 jumps to 3 and P3 back to 2

<u>P(X) (X>3)</u>

*1-2* *P(X-1) (3-2)*
*free disk X*
*P(X-1) (1-3)*

*1-3* *P(X-1) (2-3)*
*free disk X*
*P(X-1) (1-2)*

*2-1* *P(X-1) (3-1)*
*free disk X*
*P(X-1) (2-3)*

*2-3* *P(X-1) (1-3)*
*free disk X*
*P(X-1) (2-1)*

*3-1* *P(X-1) (2-1)*
*free disk X*
*P(X-1) (3-2)*

*3-2* *P(X-1) (1-2)*
*free disk X*
*P(X-1) (3-1)*

**Considerations**

This last formula looks easy and also reveals the complexity of operations, because in order to move a P(8) it will be necessary to move a P(7), free disk 8 and move again a P(7), but to move a P(7) you need to move two times a P(6) and so on.. till using two times the P(3) formulas that are on the previous page.

Here are evident **rules 1** and **3**, while the use of **rule 2** is implicit.

*by Polarsun*