

The Tower of Hanoi

From a puzzle to a proof



A legend

In a temple in Hanoi, monks move **64 golden disks** between three diamond pegs.

- They move one disk at a time, day and night.
- When the whole tower has been moved, **the world will end.**

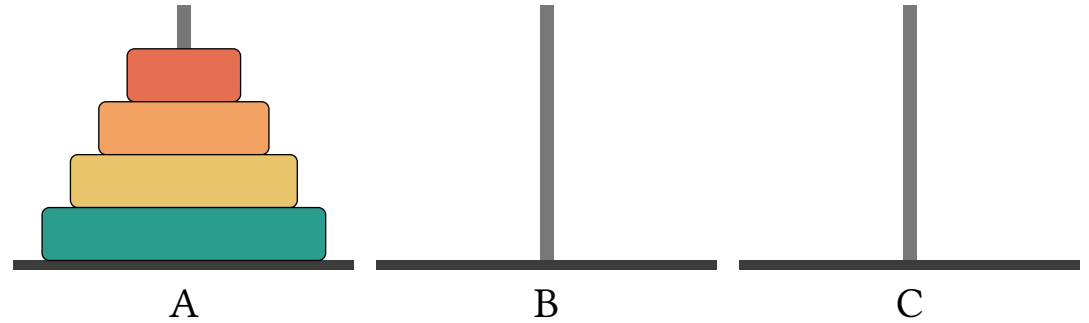
Story invented by the French mathematician Édouard Lucas in 1883.

Question: should we be worried?

The rules

Goal: move the whole tower from peg A to peg C.

1. Move **one disk** at a time.
2. Never put a **larger disk on a smaller one**.



Your turn!

In pairs, solve the puzzle with 1, 2, 3, then 4 disks.

Write down the **smallest number of moves** you managed for each.

Disks n	1	2	3	4	5
Moves					

No puzzle at hand? Play online:

<https://urbain.vaes.uk/misc/hanoi/>

or <https://www.mathsisfun.com/games/towerofhanoi.html>

Stuck? How did you do it with 2 disks? Can you reuse that?

What did you find?

Disks n	1	2	3	4	5
Moves u_n	1	3	7	15	?

Do you see a pattern?

Two conjectures

“Double it and add 1”

$$u_{n+1} = 2u_n + 1$$

$$1 \rightarrow 3 \rightarrow 7 \rightarrow 15 \rightarrow 31$$

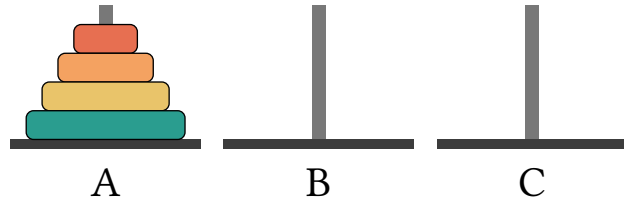
“One less than a power of 2”

$$u_n = 2^n - 1$$

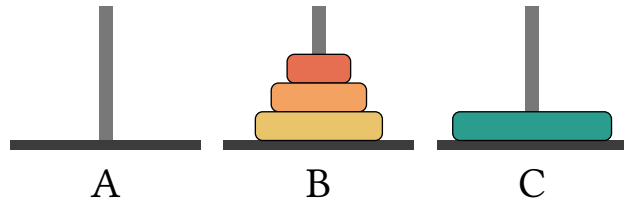
$$2 - 1, \quad 4 - 1, \quad 8 - 1, \quad 16 - 1$$

These are **guesses** based on 4 examples. Are they true for every n ?

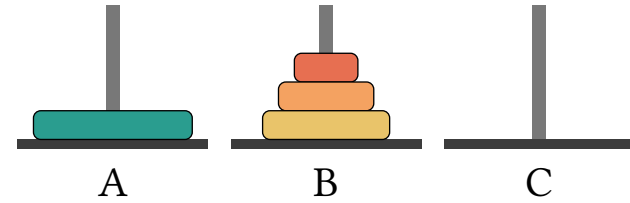
The key idea: think recursively



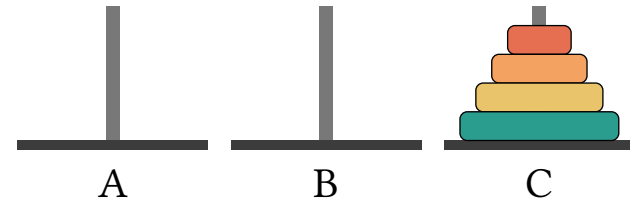
Start



2. Move the largest disk to C



1. Move the $n - 1$ top disks to B



3. Move the $n - 1$ disks onto it

Counting the moves

To move n disks:

$$u_n = \underbrace{u_{n-1}}_{\text{step 1}} + \underbrace{1}_{\text{step 2}} + \underbrace{u_{n-1}}_{\text{step 3}} = 2u_{n-1} + 1$$

Recursion: solve a problem by assuming you already know how to solve a **smaller** version of it.

Can we do better?

No! The strategy is **optimal**:

- At some point, the **largest disk** must move from A to C.
- At that moment, the $n - 1$ other disks must all be on B. $\rightarrow$ at least u_{n-1} moves before.
- Afterwards, they must all go back on top of it. $\rightarrow$ at least u_{n-1} moves after.

So no strategy can use fewer than $2u_{n-1} + 1$ moves.

Proof by induction



To make **all** the dominoes fall, two things are enough:

1. The **first** domino falls.
2. Whenever one domino falls, it knocks over the **next** one.

Proving $u_n = 2^n - 1$

Base case. $u_1 = 1 = 2^1 - 1$ ✓

Inductive step. Assume $u_n = 2^n - 1$ for some n . Then

$$u_{n+1} = 2u_n + 1 = 2(2^n - 1) + 1 = 2^{n+1} - 2 + 1 = 2^{n+1} - 1 \quad \checkmark$$

Conclusion: for every $n \geq 1$, moving n disks takes exactly $2^n - 1$ moves, and no fewer.

Should we be worried?

With 64 disks:

$$2^{64} - 1 = 18\,446\,744\,073\,709\,551\,615 \approx 1.8 \times 10^{19} \text{ moves}$$

- At 1 move per second: about **585 billion years** (roughly 40 times the age of the universe).
- A computer doing 10^9 moves per second would still need about **585 years**.

Exponential growth: one more disk means twice as much work.

Hanoi and binary numbers

Write the move number k in binary. The disk that moves is given by the position of the **rightmost 1**.

Move k	1	2	3	4	5	6	7
Binary	00 1	0 1 0	01 1	1 00	10 1	1 1 0	11 1
Disk	1	2	1	3	1	2	1

Solving the puzzle with n disks is the same as **counting from 1 to $2^n - 1$** .

A rule without recursion

1. Move the **smallest disk** one peg along, always in the **same direction** ($A \rightarrow B \rightarrow C \rightarrow A \rightarrow \dots$).
2. Make the only legal move that **does not** use the smallest disk.
3. Repeat.

This gives exactly the same $2^n - 1$ moves.

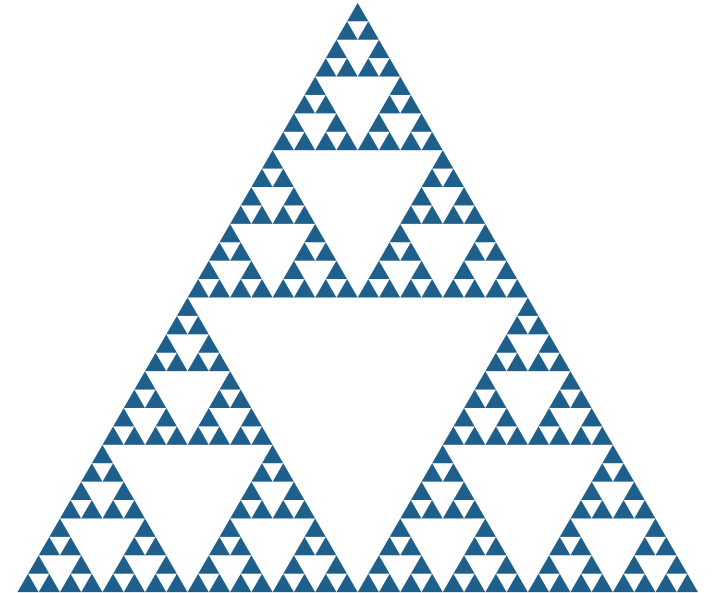
(With an odd number of disks, go $A \rightarrow C \rightarrow B \rightarrow A$ instead, to finish on C.)

A fractal hidden in the puzzle

- Draw a point for every possible position (3^n of them).
- Connect two positions if one move goes from one to the other.

You get the **Sierpiński triangle**!

The optimal solution goes straight along one side, from one corner to another.



A problem still open today

What if there are **4 pegs** or more?

- Frame and Stewart proposed their strategy in 1941.
- For 4 pegs, it was only **proved optimal in 2014** (Thierry Bousch).
- For 5 pegs or more, **nobody knows** whether it is optimal!

A puzzle you can explain to a child can lead to problems that are still unsolved.

Your turn: 4 pegs

With **4 pegs**, how few moves do you need?

Disks	3	4	5
3 pegs	7	15	31
4 pegs			

Play online (choose 4 pegs):

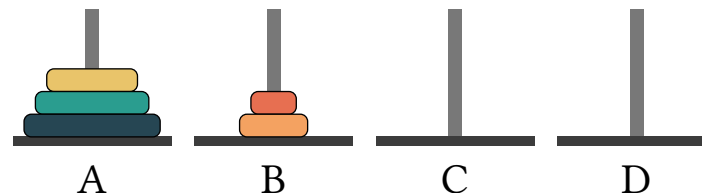
<https://urbain.vaes.uk/misc/hanoi/>

or <https://towersofhanoi.info/Play.aspx> (then click “Restart”)

Hint: start by putting a few small disks aside.

Frame–Stewart: a strategy for 4 pegs

1. Move the k top disks to B, using **all 4 pegs**.
2. Move the $n - k$ others to D with **only 3 pegs**:
 $2^{n-k} - 1$ moves.
3. Move the k disks onto D, using **all 4 pegs**.



After step 1 ($n = 5, k = 2$)

$$F_n = \min_k (2F_k + 2^{n-k} - 1)$$

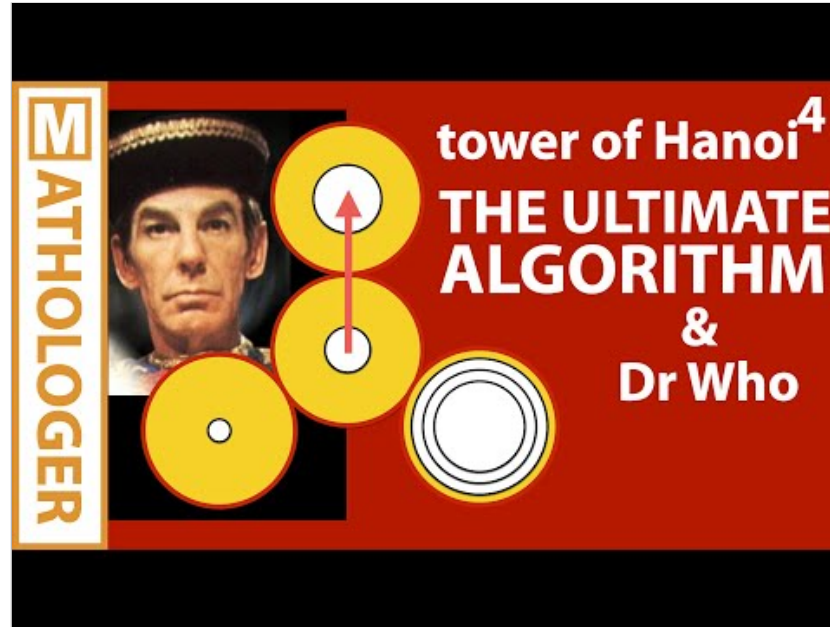
Disks	1	2	3	4	5	6	7	8	64
3 pegs	1	3	7	15	31	63	127	255	$\approx 1.8 \times 10^{19}$
4 pegs	1	3	5	9	13	17	25	33	18 433

What we learned

1. **Experiment** and look for patterns.
2. Make **conjectures**.
3. Think **recursively**: reduce a problem to a smaller one.
4. **Prove** the result with induction.
5. **Exponential growth** gets huge very fast.

Challenge: solve it with 5 disks in exactly 31 moves!

To go further



Mathologer — “The ultimate tower of Hanoi algorithm”

<https://www.youtube.com/watch?v=MbonokcLbNo>

What happens with 4 pegs or more, surprising Hanoi constants, and much more.