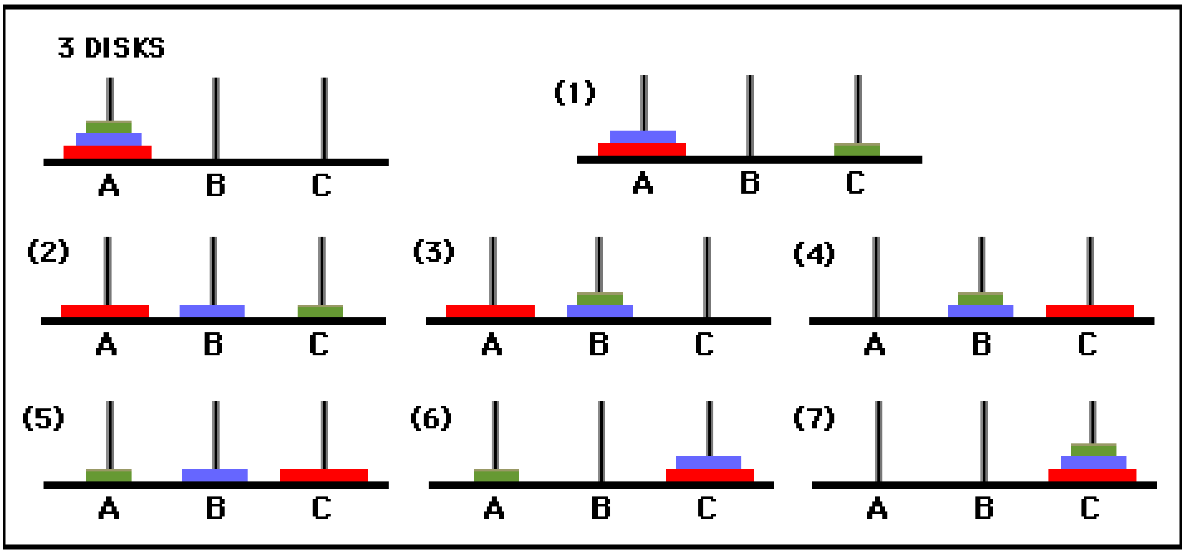
# The Tower of Hanoi

The Legend goes, that in the temple of Benares, beneath the dome which marks the centre of the world, there lies a brass plate with three diamond needles fixed to it. At the beginning of time God placed 64 disks of pure gold upon the first needle, largest resting on the brass plate, with each disk progressively smaller until reaching the top. He then ordained that a group of priests should transfer them to the third needle and declared that they must work in accordance with the unchangeable laws of Brahma. The first law states that 'Thou shall not move more than one disk at a time'. The second law states that 'no disk should ever rest upon a disk smaller than itself'. The priests reportedly worked day and night at their task. It is said that, when they finish, the Tower will crumble and the world will end.

This grand story in fact accompanies a toy puzzle, the Tower of Hanoi, invented by Edouard Lucas, a French mathematician in 1883. Luckily for people trying to solve the puzzle, the tower of Hanoi is a little smaller than the Tower of Brahma, involving only 8 disks instead of 64. One might be confused at first, it's not immediately obvious that the puzzle has a solution, just move the tower from one peg to another in accordance to the rules, what's the challenge? Well after a small go at the puzzle or with a little thought, the question arises: what is the best way to do this? What's the smallest number of moves required to solve the puzzle for n disks?

Instead of focusing on how many moves it will specifically take for the Tower of Hanoi's 8 disks or the tower of Brahma's 64 disks – generalization helps, as it allows you to scale down the problem.  This is part of the reason I find this problem so interesting; it is a great example of generalizing to spot patterns and recurrences; consequently, it serves as a useful introduction to the mathematics of computer science and algorithms.

The first step to solving this problem is to establish appropriate notation: $T_n$ will be the minimum number of moves required to transfer n disks, following Lucas's rules. Thus, for the small values of n we can state: $T_0=0$, $T_1=1$ and $T_2=3$.



Looking more closely at how to solve for three disks, gives some clues on how to solve on a larger scale. First get the smallest two disks off the largest disk (requiring $T_2$ moves), then transfer the largest disk to the other peg (requiring one move) and lastly transfer the two smaller disks back onto the larger one (requiring another $T_2$ moves). This process can be quite easily generalized. Move $T_{n-1}$ disks to another peg, transfer the largest disk, then transfer $T_{n-1}$ disks back onto the large one. Hence, transferring n disks requires $2T_{n-1}+1$ moves:

$T_n = 2T_{n-1} + 1$ for $n > 0$

$T_0 = 0$This recurrence equation can compute for any value of n. However, for large values of n it would take a long time to work out – so a nice, neat closed form equation for $T_n$ would be more useful. Looking at the first few values for $T_n$ could help:

$T_3 = 2 \times 3 + 1 = 7$

$T_4 = 2 \times 7 + 1 = 15$

$T_5 = 2 \times 15 + 1 = 31$

$T_6 = 2 \times 31 + 1 = 63$

$T_7 = 2 \times 6$... Bingo! There seems to be a pattern:

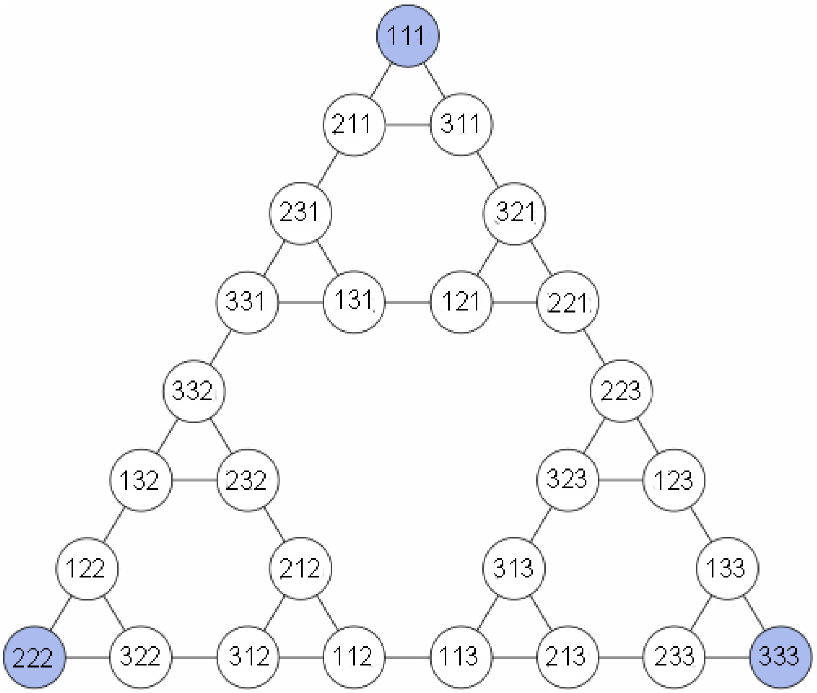
$T_n = 2^n - 1$ for $n \geq 0$

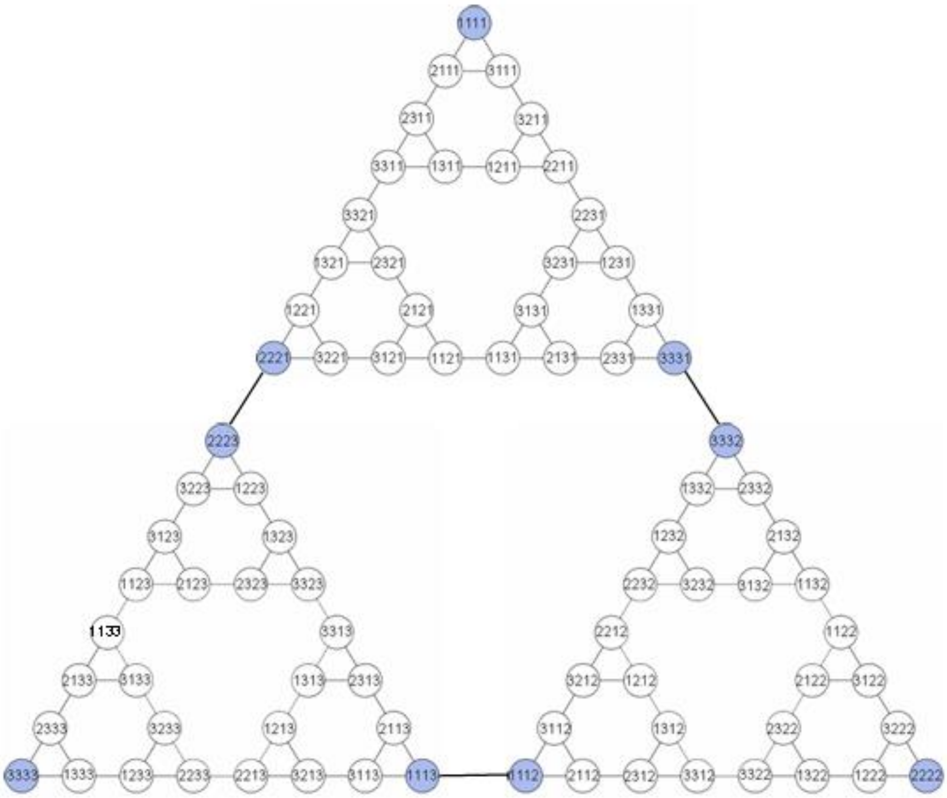
Lastly, the statement needs to be checked to ensure that it is true for all values of n, as knowing it is true for the first 6 values, while a good indication, doesn't prove it is true for $n \geq n_0$. This part is pretty simple, as it just involves substituting the new equation into one we already know is true for all values of n.

$T_n = 2T_{n-1} + 1 = 2(2^{n-1} – 1) + 1 = 2^n – 1$

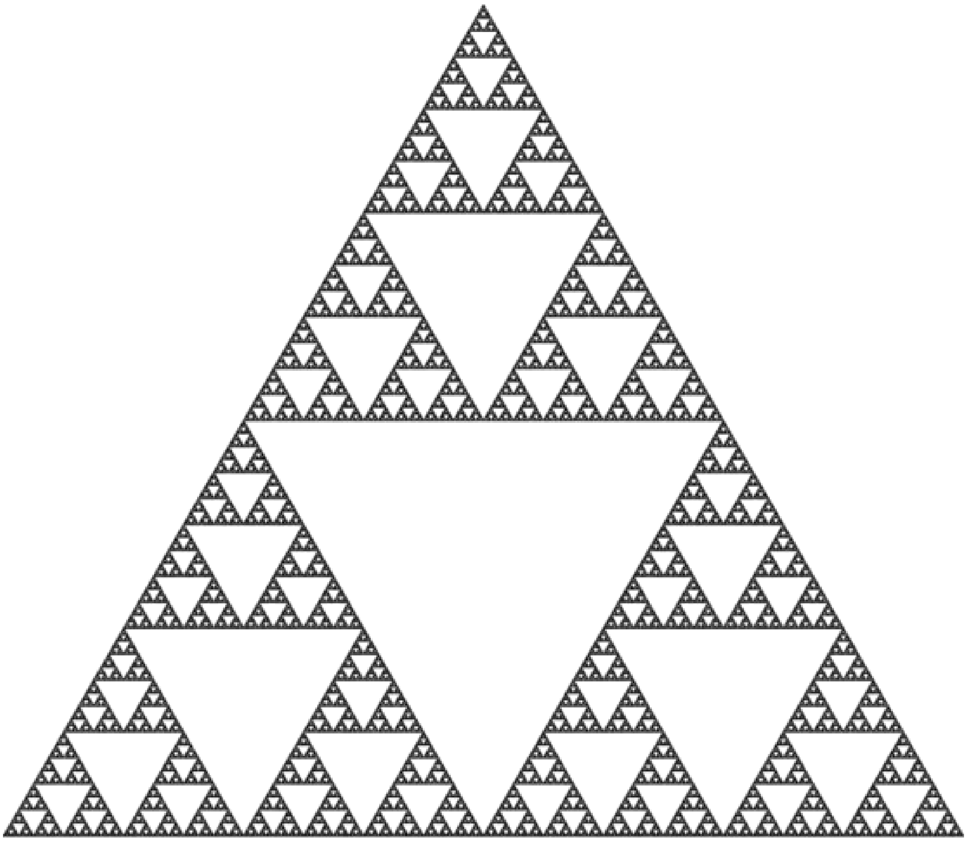
Proved.

What's particularly fascinating to me about this puzzle is its connection to Sierpinski's triangle. When graphing all the possible configurations of a game of 3 disks and 3 pegs you get a sort of 'triangle network', which is officially called a Hanoi graph. The positions are encoded using a triple. For example, disk one and disk 3 on peg 2, with disk 2 on peg 3 will look like (2, 3, 2).





As you consider more disks, the pattern continues, for instance here is a graph representing 4 disks. Following this notion of increasing the number of disks, an intriguing question emerges. What would the graph for an infinite number of disks look like? Well, regard the image below:



This is a fractal, known as Sierpinski's triangle. It can be generated from another infinite process. Step 1 is to draw an equilateral triangle. Step 2 is to divide the equilateral triangle into 4 smaller congruent equilateral triangles, discarding the central triangle. Step 3 is to repeat step 2 in the smaller triangles infinitely many times.

Whilst being quite beautiful and interesting, one still might ask: what is the purpose of these connections? How are they useful? Firstly, let's entertain this question: suppose you travel around Sierpinski's triangle without ever leaving the fractal, what is the average distance between two points? Well, a lot of times in mathematics problems are like boulders, blocking the middle of the road. At first you might try to push the boulder or dig through it – facing the problem head on. However, in many cases stepping back from the problem, looking at its connections to other areas of mathematics provides an easier route to your answer – you could just climb over the boulder. Convoluted metaphor aside, no one had been able to solve this problem until Hinz used Hanoi graphs to figure out the average distance is 466/885 (given the length of one side of the triangle is 1).

Now, turning attention back to the poor priests, who are of course still dutifully moving disks, and will be for very long time as for n=64 there are around 18 quintillion moves ($2^{64} – 1$). Even at the rate of one move per second it will still take the priests 5.85 quadrillion years to complete. All in all, it's safe to assume that the temple has some years of life left, as does the world.