

Classic Riddler

5 February 2021

Riddle:

Cassius the ape (a friend of Caesar's) has gotten his hands on a Lucas' Tower puzzle (also commonly referred to as the "Tower of Hanoi"). This particular puzzle consists of three poles and three disks, all of which start on the same pole. The three disks have different diameters—the biggest disk is at the bottom and the smallest disk is at the top. The goal is to move all three disks from one pole to any other pole, one at a time, but there's a catch. At no point can a larger disk ever sit atop a smaller disk.

For N disks, the minimum number of moves is $2^N - 1$. (Spoiler alert! If you haven't proven this before, give it a shot. It's an excellent exercise in mathematical induction.)

But this week, the *minimum* number of moves is not in question. It turns out that Cassius couldn't care less about solving the puzzle, but he is very good at following directions and understands a larger disk can never sit atop a smaller disk. With each move, he randomly chooses one among the set of valid moves.

On average, how many moves will it take for Cassius to solve this puzzle with three disks?

Extra credit: On average, how many moves will it take for Cassius to solve this puzzle in the general case of N disks?

Solution:

I set up the script `Towers_of_Hanoi.C` to solve this riddle. It solves this game for an arbitrary number of discs, although for practicality, I have stopped at seven discs.

The script creates a vector with one entry for each disc, and the value of each entry (1, 2, or 3) represents the peg position of that disc. For N discs, there are 3^N possible vectors, and each of these vectors corresponds to exactly one legal arrangement of the discs on the pegs. Specifically, each arrangement can be made by placing the discs from largest to smallest according to the order of the peg numbers in the vector.

From the initial arrangement (a vector of all 1s), there are only two legal moves; the smallest peg can be moved to either of the two empty pegs. For all other arrangements (except all 2s or all 3s, which are winning arrangements), there are exactly three legal moves. The smallest peg can still be moved to either of the other two pegs, or a larger disc can be moved. This third move is forced because it cannot be moved to the smallest disc's peg, and only the smallest remaining peg could be moved, anyway.

Based on a random number generator, the discs are moved among the legal moves, with a probability of $1/2$ for the initial move (or any subsequent move from the initial arrangement), and a probability of $1/3$ for the other moves. The script counts the moves being made then calculates and displays the average for each number of discs. I decided to run 1×10^6 simulations for each number of discs, to get the answers (hopefully) reasonably converged. It could probably handle more discs with fewer simulations.

For three discs, the script calculated an average of **70.7 moves**.

Here are my other averages so far:

Discs	Average moves
1	1 (trivial)
2	10.6
3	70.7
4	404
5	2150
6	11155
7	56985

Of course, they are increasing faster than 2^N , but I don't have enough points to examine the long-term behavior. Nor do I have the mathematical skills to find an analytical answer. From what I have, it looks like they are increasing around 5^N , but not quite. Perhaps they eventually approach $4^N = 2^{2N}$, which sounds nice, but I can't prove it.

Addendum

I have since set up a series of linear equations to determine the expected number of moves remaining from a given arrangement. I solved this with a 27×28 matrix, which can be seen in `Tower_matrix_3.mat`. The exact solution is therefore $637/9 \approx 70.778$.