Example – Towers of Hanoi

The Towers of Hanoi is a mathematical game or puzzle. It consists of three pegs, and a number of disks of different sizes which can be put onto a peg. The puzzle starts with the disks in a stack in order from smallest to largest on one peg, smallest at the top. The objective is to move the entire stack from one peg to another peg, obeying the following rules:

- Only one disk may be moved at a time
- One move consists of taking the top disk from one of the pegs and putting it onto another peg, on top of the other disks on that peg
- No disk may be placed on top of a smaller sized disk

The puzzle was invented by the French mathematician Édouard Lucas under the name N. Lucas de Siam in 1883. There is a legend about a temple in Benares with a dome which marked the center of the world. Within the dome, there is a large room with three posts in it surrounded by 64 golden disks. The priests of Hanoi, move these disks, in accordance with the rules of the puzzle. According to the legend, when the last move of the puzzle is completed, the universe will come to an end. (There are many variations on this legend.)

If the legend were true, and if the priests were able to move disks at a rate of one per second, using the smallest number of moves, it would take them $2^{64}-1$ seconds or roughly 585 billion years to finish.

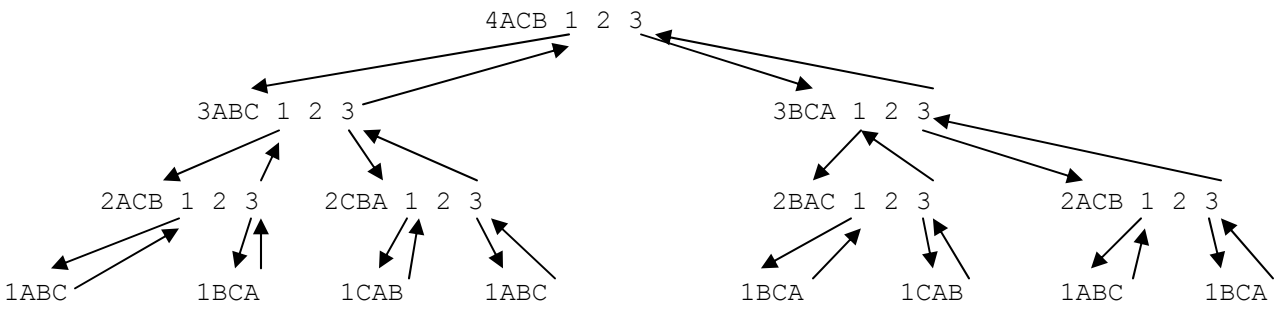
Solve recursion:
Step 1: moveDisks – **will** move any number of disks from a peg to another peg using extra peg as an auxiliary

Step 2: base case – one disk, easy, just move it

Step 3: recursive part – if you have a function that will move any number of disks from one to another,
Move n-1 disks from the peg they are initially on, the *from peg*, to the *auxiliary peg*
Then move the n^{th} disk (largest disk) to the peg you want to ultimately move them to, the to peg
Move n-1 disks sitting on the *auxiliary peg* to the peg you want to ultimately move them to, *to peg*

Step 4: code

Draw an execution tree (showing the calls and returns) of the execution of movedisks(4, 'A', 'C', 'B') and display output:



Output:

Move Disk #	From Peg	To Peg
1	A	B
2	A	C
1	B	C
3	A	B
1	C	A
2	C	B
1	A	B
4	A	C
1	B	C
2	B	A
1	C	A
3	B	C
1	A	B
2	A	C
1	B	C