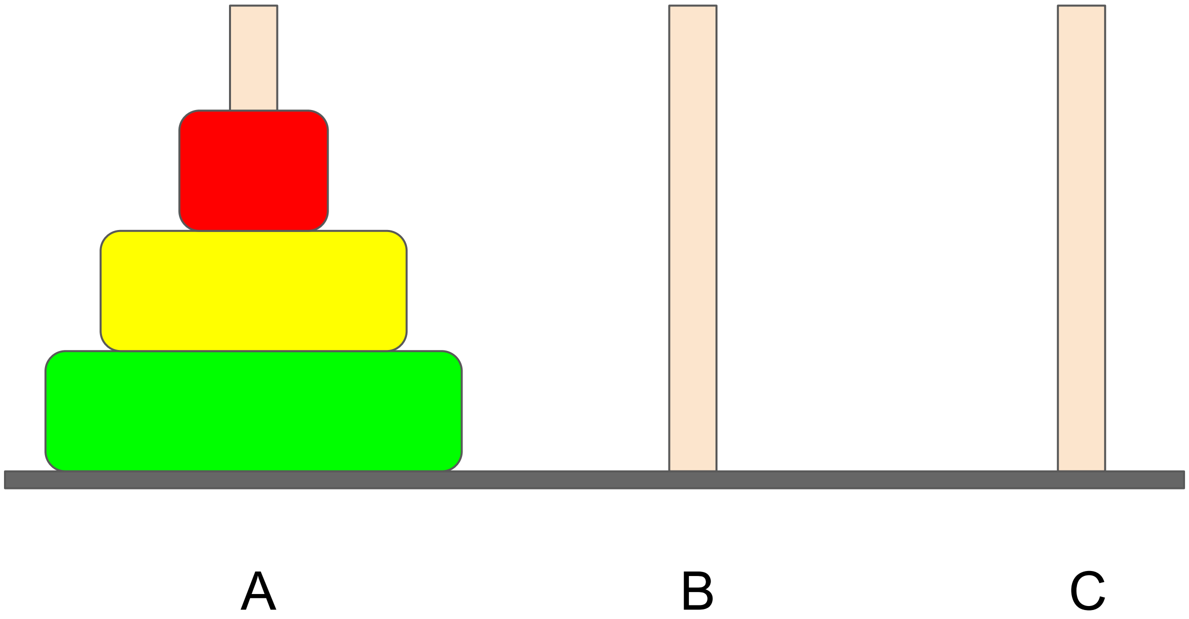
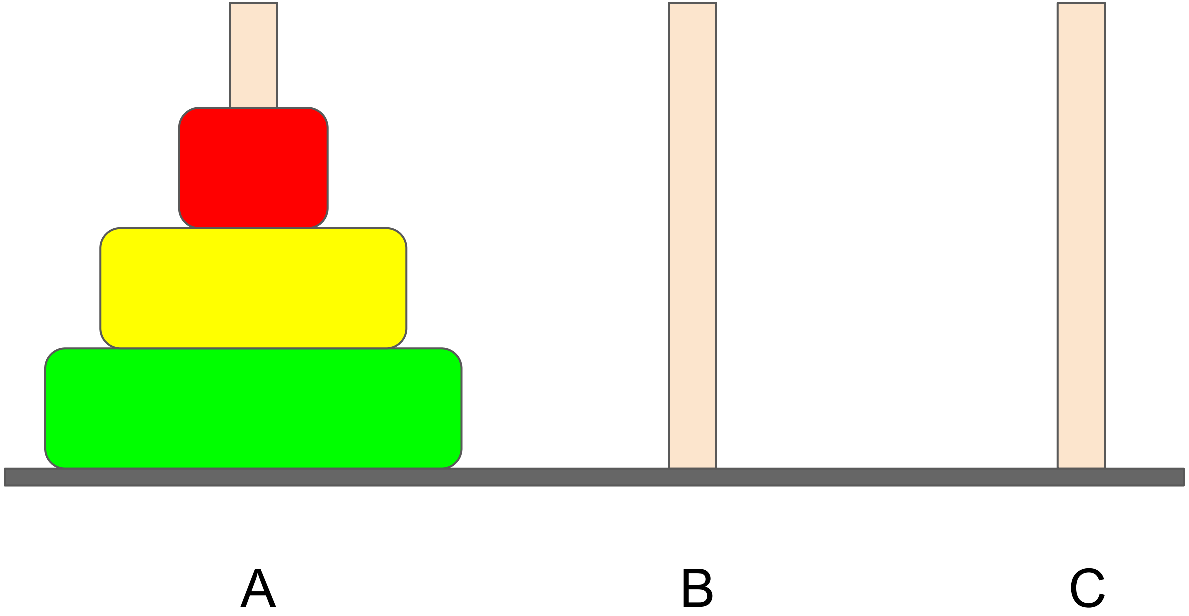
Tower of Hanoi



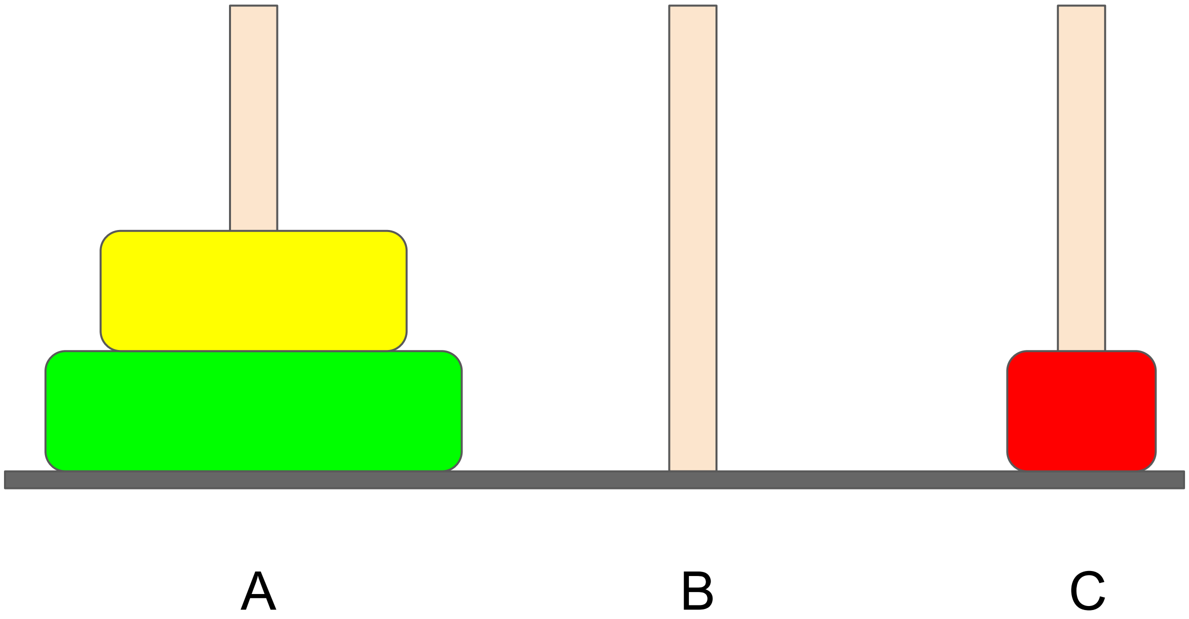
Rules

- Can only move the top disk from each peg.
- Can only move a disk to the top of another peg.
- Can only move 1 disk at a time.
- There are 3 pegs, A, B, and C, with all the disks stacked on A to begin with.
- Objective is to get all rings to the rightmost peg, C.

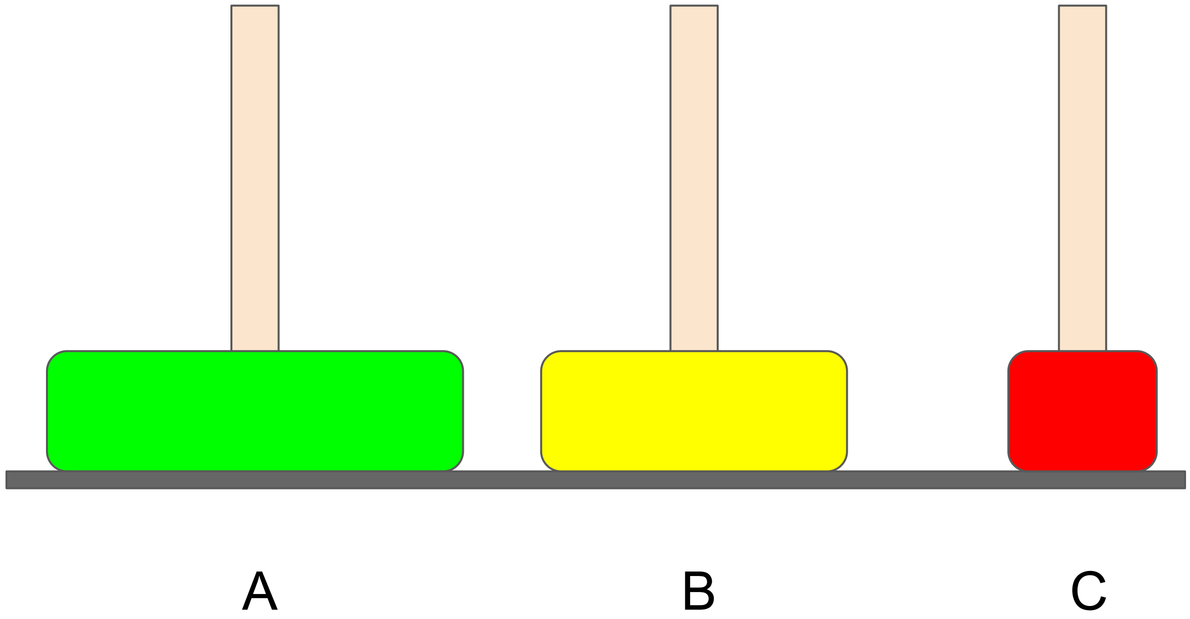
Optimal Solution(with n = 3 disks):



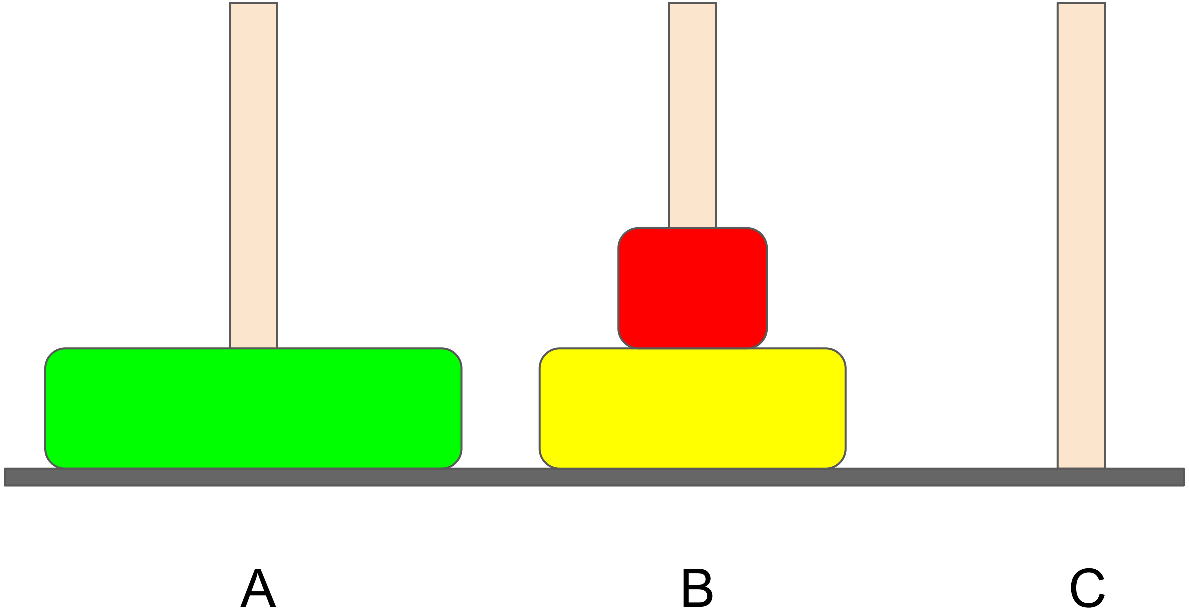
Optimal Solution:



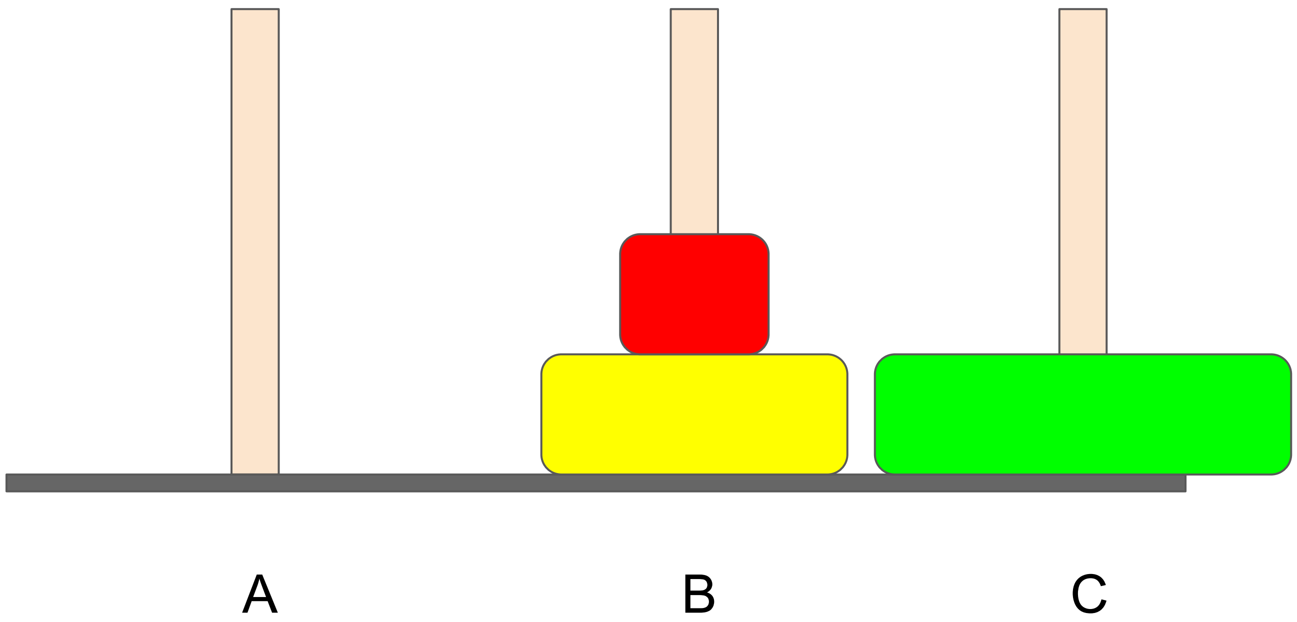
Optimal Solution:



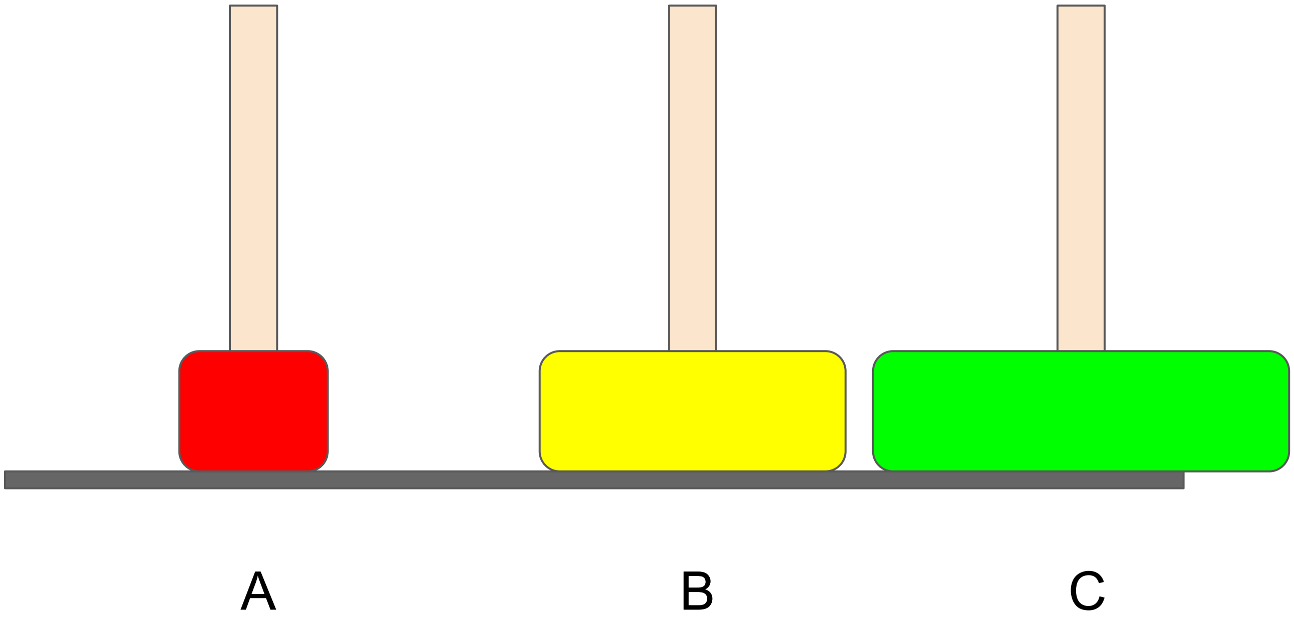
Optimal Solution:



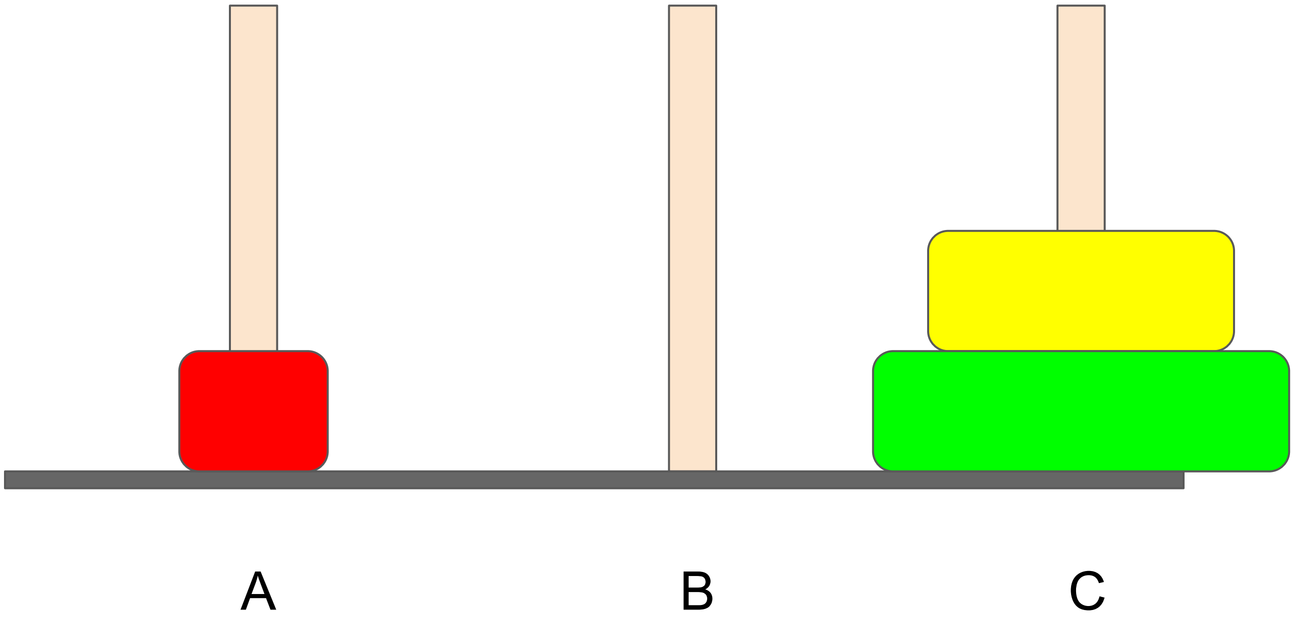
Optimal Solution:



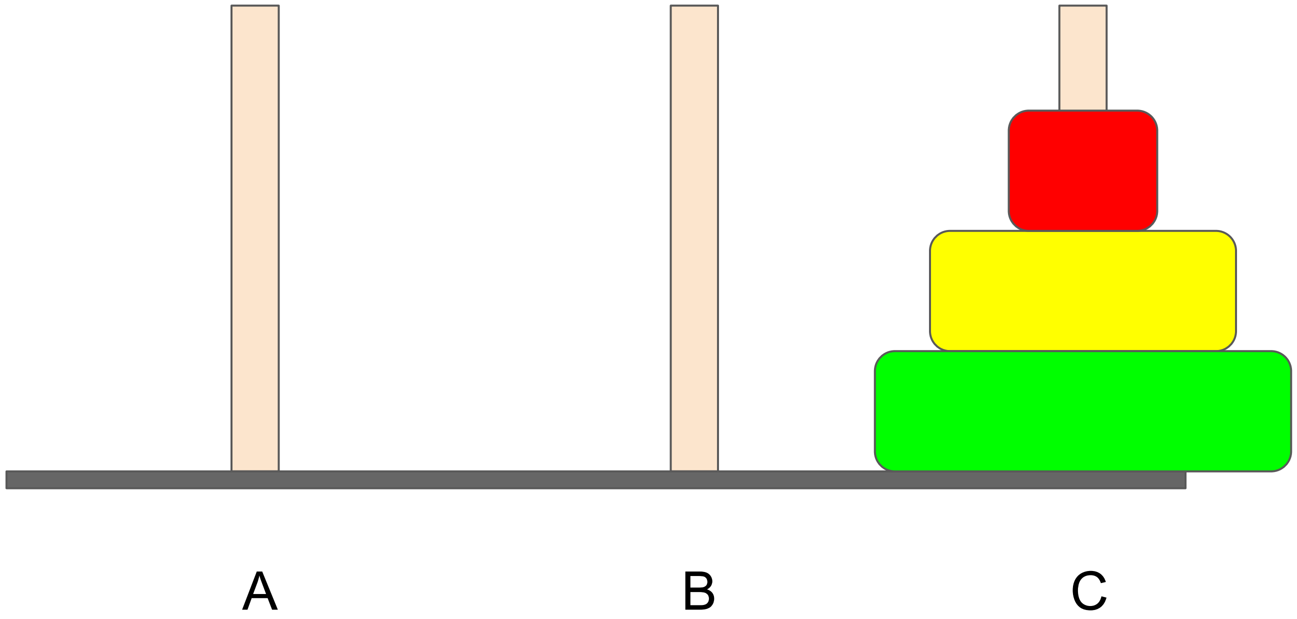
Optimal Solution:



Optimal Solution:



Optimal Solution:



The idea

Like we saw in the previous example, our first few steps brought us so that the smaller two rings are on the peg B and the largest on peg C. To do this, we first had to move the smallest to peg C and then the middle to peg B. After the two smaller rings reached peg B, we moved them over to C, by moving the smallest to A, the middle to C, and finally, the smallest to C.

Recursive Algorithm

Now we can break this problem up into smaller subproblems. To solve the problem as a whole, we want to move n - 1 disks to the middle peg(which we will call the auxiliary peg), and then move the last ring to the target, and finally move our n - 1 rings to the target. We can define each subproblem as moving the n - 1 rings from the target to the auxiliary, and we can keep recursively calling this problem to construct a complete solution while the number of rings(n) is greater than 0.