**Q 8.6** (pg. 135 *Cracking the Coding Interview, 6th Edition*) *Hard*

In the classic problem of the Towers of Hanoi, you have 3 towers and N disks of different sizes which can slide onto any tower. The puzzle starts with disks sorted in ascending order of size from top to bottom (i.e., each disk sits on top of an even larger one). You have the following constraints:

1. Only one disk can be moved at a time
2. A disk is slid off the top of one tower and onto another tower
3. A disk cannot be placed on top of a smaller disk

Write a program to move the disks from the first tower to the last using stacks.

*Hint 1*: Try the Base Case and Build approach

*Hint 2*: You can easily move the smallest disk from one tower to another. It’s also pretty easy to move the smallest two disks from one tower to another. Can you move the smallest three disks?

*Hint 3*: Think about moving the smallest disk from tower X=0 to tower Y=2 using tower Z=1 as a temporary holding spot as having a solution for f(1, X=0, Y=2, Z=1). Moving the smallest to disks is f(2, X=0, Y=2, Z=1). Given that you have a solution for f(1, X=0, Y=2, Z=1) and f(2, X=0, Y=2, Z=1), can you solve f(3, X=0, Y=2, Z=1)?

*Hint 4*: Observe that it doesn’t really matter which tower is the source, destination, or buffer. You can do f(3, X=0, Y=2, Z=1) by first doing f(2, X=0, Y=1, Z=2) (moving two disks from tower 0 to tower 1, using tower 2 as a buffer) then moving disk 3 from tower 0 to tower 2, then doing f(2, X=1, Y=2, Z=0) (moving tow disks from tower 1 to tower 2, using tower 0 as a buffer). How does this process repeat?

*Hint 5*: If you’re having trouble with recursion, then try trusting the recursive process more. Once you’ve figured out how to move the top two disks from tower 0 to tower 2, trust that you have this working. When you need to move three disks, trust that you can move two disks from tower to another. Now, two disks have been moved. What do you do about the third?