Problem
=======
Jennifer Widow

Towers of Hanoi

There are three posts, P_1, P_2, and P_3. Post P_1 starts with a tower of N disks on it, D_1, D_2, ..., D_N, of strictly decreasing size from bottom to top. The goal is to move all N disks from post P_1 to post P_2, possibly via post P_3, subject to:

(1) At most one disk may be moved at a time.
(2) No disk may ever be placed on top of a smaller disk.

Specifically, write a general procedure:

Move({disk1, disk2, ..., diskM}, post1, post2, post3)

that emits a sequence of instructions for moving disk1, disk2, ..., diskM from post1 to post2, possibly via post3. To solve the original problem we call:

Move({D_1, D_2, ..., D_N}, P_1, P_2, P_3)

Each emitted instruction is of the form "move D_i from P_j to P_k".

Hint on request:

Use recursion-- note that Move() can be called with any set of disks and any parameter ordering of the three posts.

Additional/alternate problems:

(A1) Write a function that takes an argument N and returns the number of moves required to solve the problem with N disks.

(A2) What is the computational complexity of the problem (in #disks)?

Solution

```
Move({disk1, disk2, ..., diskM}, post1, post2, post3):
  If M=1 then emit "move [disk1] from [post1] to [post2]"
  Else
    Move({disk2, disk3, ..., diskM}, post1, post3, post2)
    Move({disk1}, post1, post2, post3)
    Move({disk2, disk3, ..., diskM}, post3, post2, post1)
```

(A1) f(1) = 1; f(N > 1) = 2*f(N-1) + 1

(A2) Exponential in N: O(2^N) Specifically, f(N) = 2^N - 1