How do we solve problems recursively?

Recursive Solutions: Fibonacci Numbers.

- Identify the base case:

  When `n = 0` or `n = 1`

  `fib(0)` or `fib(1) = 1`

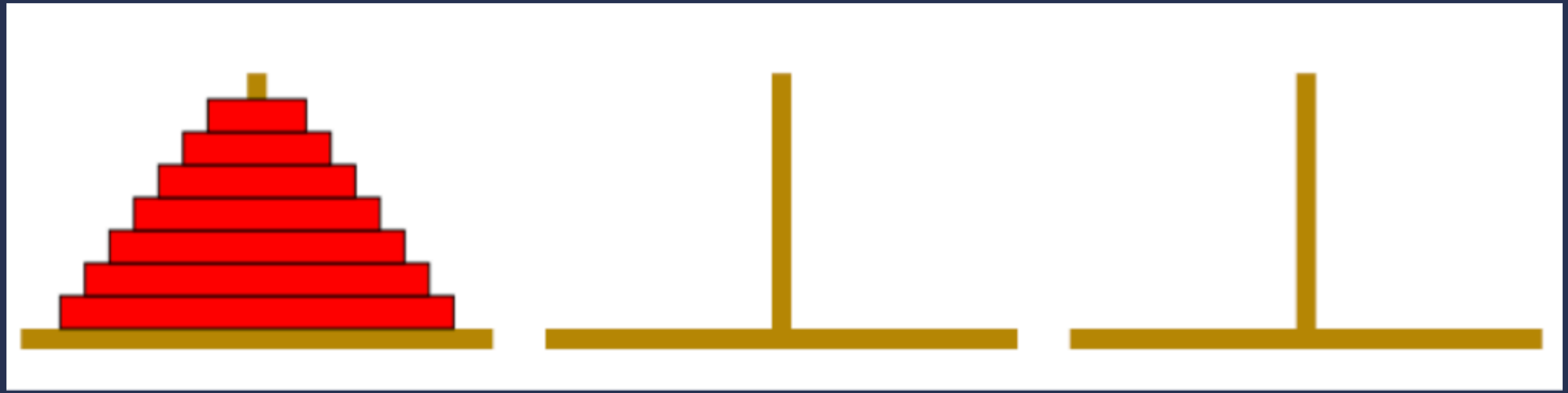
1. `fib(n)` relies on `fib(n-1)` and `fib(n-2)`

2. `fib(n) = fib(n-1) + fib(n-2)`

Code:

```
def fib(n):
    if (n < 2):
        return 1
    else:
        return fib(n-1) + fib(n -2)
```

How do we solve problems recursively?



Deep in the heart of Vietnam there is a monastery with 3 sacred pillars and 64 sacred golden disks. The monks of this monastery dedicate their lives, and the entire purpose of their order, to moving the 64 sacred disks from the start pillar to the end pillar, following the 2 divine rules:

0. Only 1 disk can be moved at a time
1. A disk can never be placed on top of a smaller disk (or a disk can never have a larger disk on top of itself.

When the monks have moved all the disks, then the entire universe will reach enlightenment.