Mathematical Analysis

- Let’s continue Recursive Algorithms

Fibonacci numbers

The Fibonacci numbers:

0, 1, 1, 2, 3, 5, 8, 13, 21, …

The Fibonacci recurrence:

$F(n) = F(n-1) + F(n-2)$

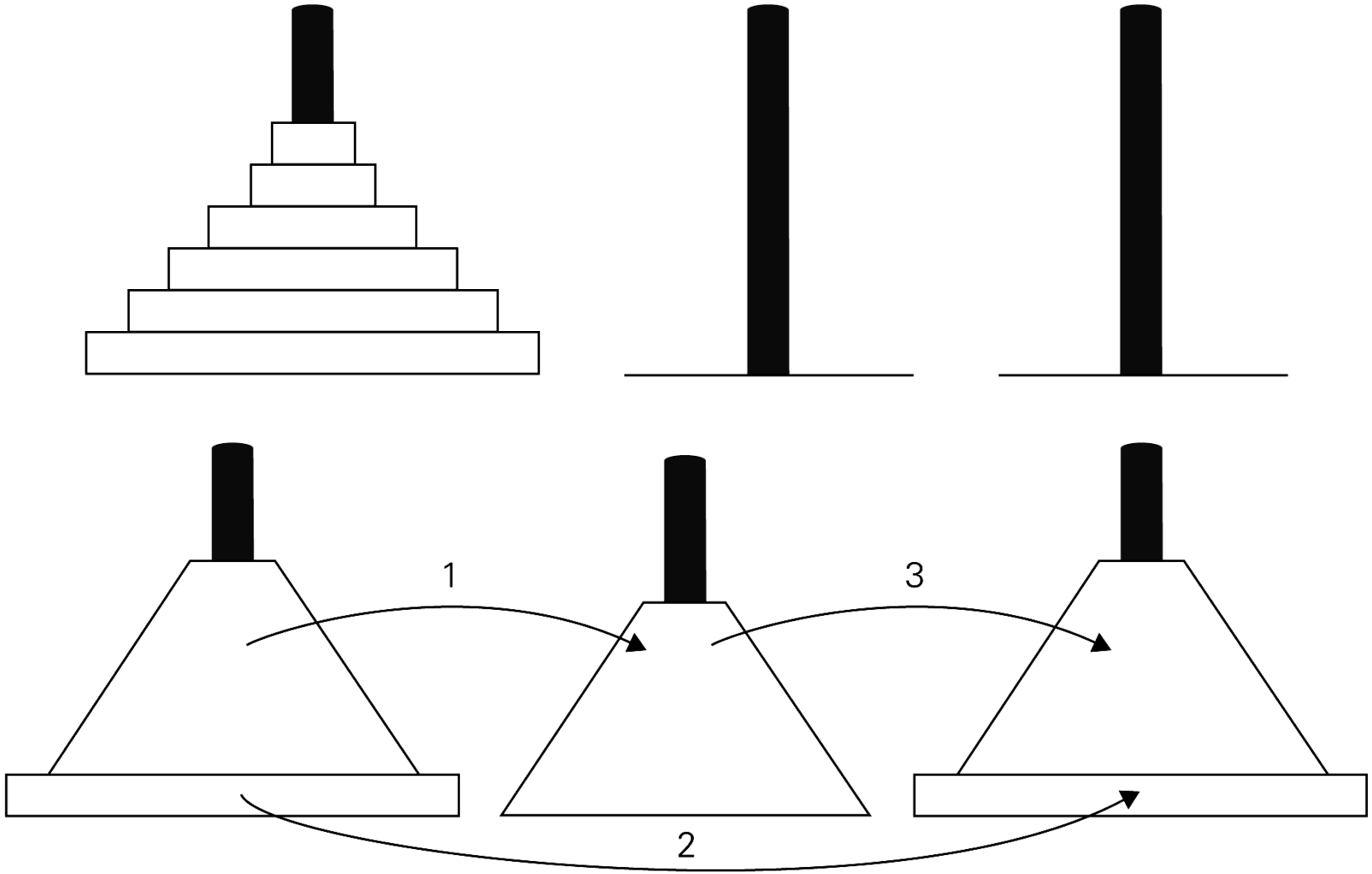
$F(0) = 0$

$F(1) = 1$

General 2nd order linear homogeneous recurrence with constant coefficients:

$$aX(n) + bX(n-1) + cX(n-2) = 0$$

Example 2: The Tower of Hanoi Puzzle



Recursive solution to the Tower of Hanoi puzzle

Example 2: The Tower of Hanoi Puzzle

Number of moves M(n) depends upon the number of disks n only. The recurrence is given by

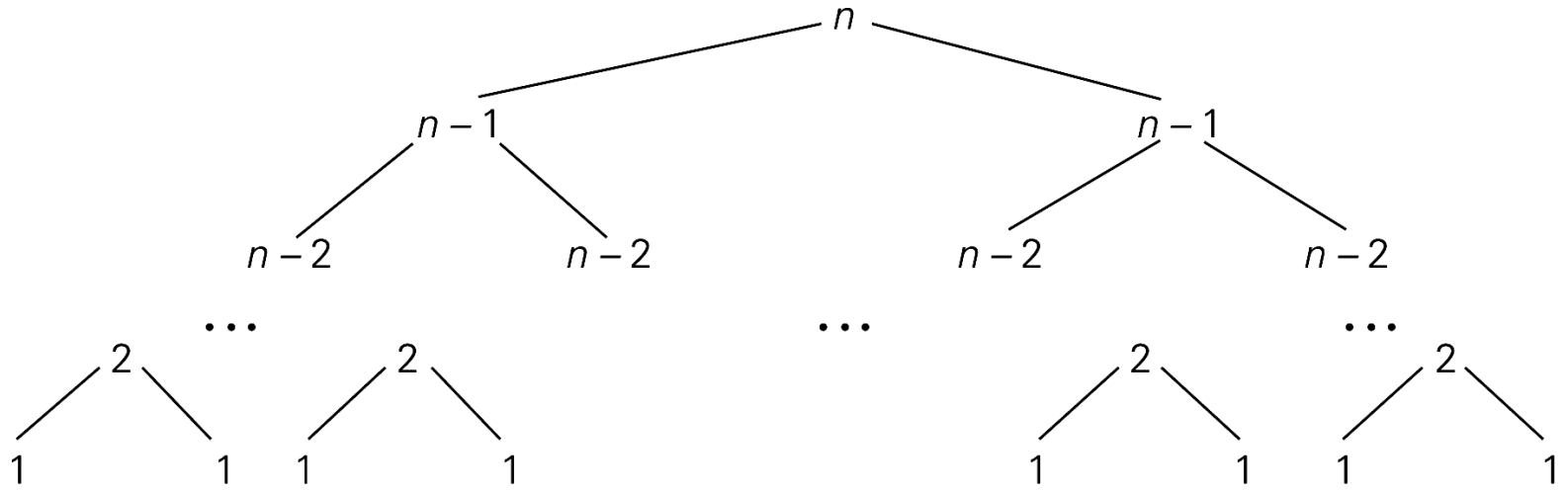
$$M(n) = M(n\text{-}1) + 1 + M(n\text{-}1) \quad \text{for } n > 1$$

The initial condition is given by M(1) = 1, thus the recurrence relation is given by

$$M(n) = 2M(n\text{-}1) + 1 \quad \text{for } n > 1$$
$$M(1) = 1$$

Now solve the recurrence relation

Tree of calls for the Tower of Hanoi Puzzle



Tree of recursive calls made by the recursive algorithm for the Tower of Hanoi puzzle

What is the total number of moves to be performed?

What is the number of nodes in the tree?

Tree of calls for the Tower of Hanoi Puzzle

Number of recursive calls with *l* levels is given by

$$C_{tree}(n) = \sum_{l=0}^{n-1} 2^l = 2^n - 1$$

which agrees with the number of moves computed above.