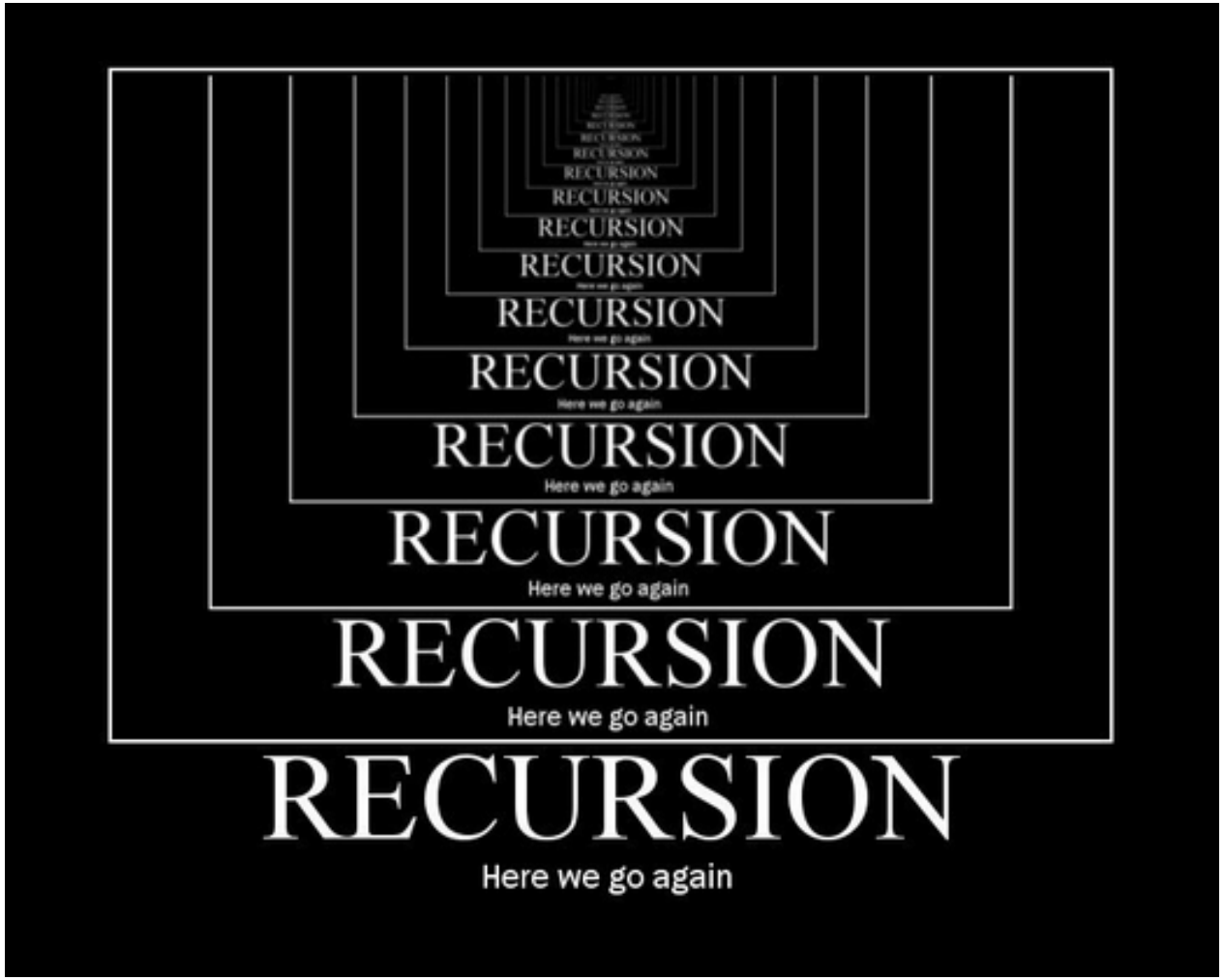
CSCI 2824: Discrete Structures

Lecture 22: Recursion

Rachel Cox

Department of Computer Science

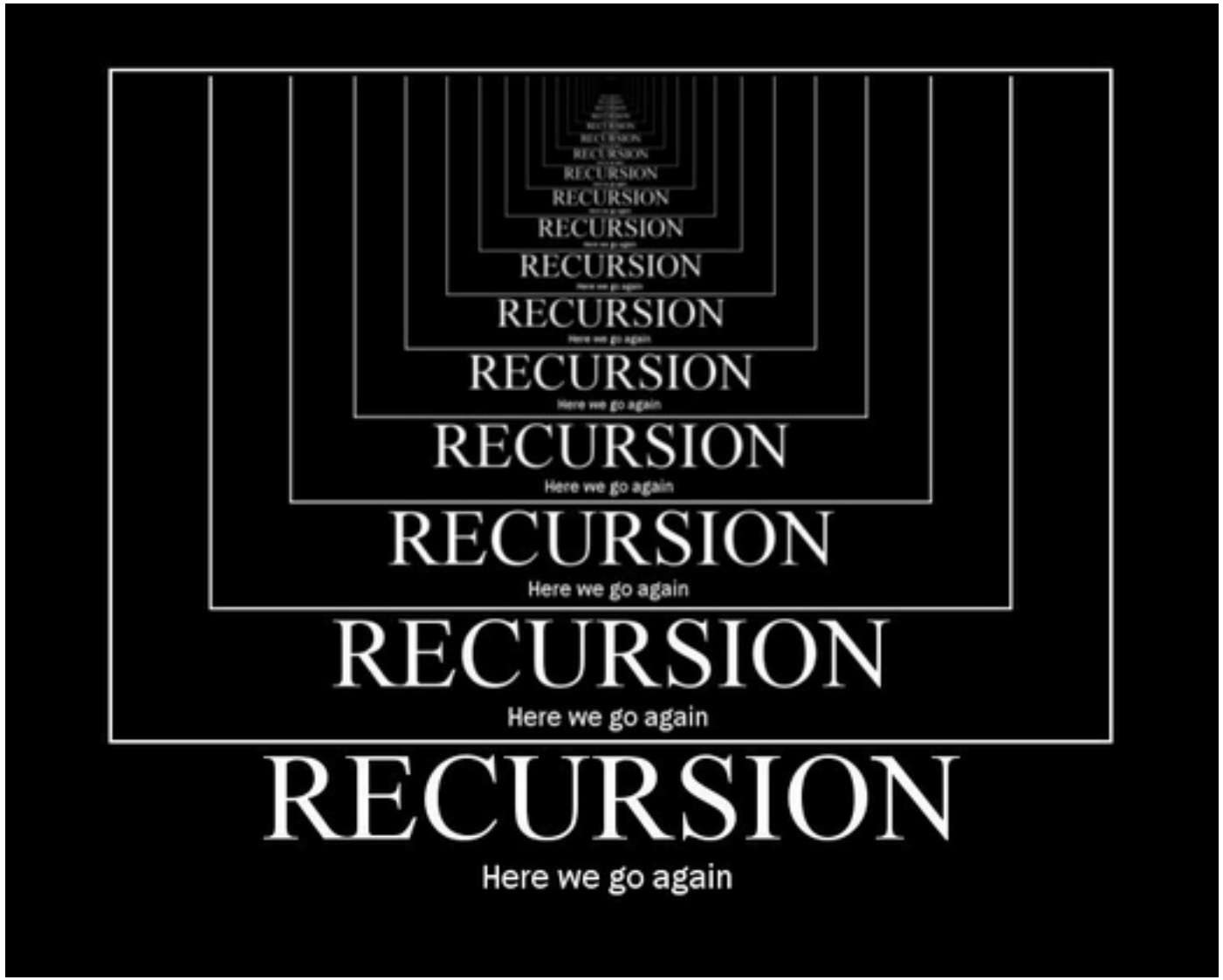


Announcements

Homework 8 - Due Friday at noon to Gradescope.

Thursday office Hours: 1:30 - 3:30 pm in ECOT 732

Recursion



Sometimes it is difficult to define a discrete object explicitly. But it might be easy to define this object in terms of itself.

This is the idea behind **recursion**.

- Recursion, Recurrence Relations, and Induction are *very* intimately related.
- Functions and Sets can be defined recursively.
- Induction lets us prove properties of Recursive algorithms.

Recursion

Example: Given the function f below, find $f(1), f(2), f(3)$, and $f(4)$.

- $f(0) = 1$
- $f(n+1) = 3f(n) - 1$

$f(1) = 3f(0) - 1$
$= 3 \cdot 1 - 1$
$= 2$

$f(2) = 3f(1) - 1$
$= 3 \cdot 2 - 1$
$= 5$

$f(3) = 3f(2) - 1$
$= 3 \cdot 5 - 1$
$= 14$

$f(4) = 3f(3) - 1$
$= 3 \cdot 14 - 1$
$= 41$

Recursion

Notice the similarity to induction. We have a base case and a rule that takes us from one value to the next.

Base case: $n = 0,\ f(0) = 1$
Recursive Step: Given $f(n-1)$ we can compute $f(n)$

Example: Find a recursive definition for $f(n) = n!$

Initial condition $f(0) = 0! = 1$

$f(1) = 1!$
$f(2) = 2! = 2 \cdot 1! = 2f(1)$
$f(3) = 3! = 3 \cdot 2! = 3f(2)$
$f(4) = 4! = 4 \cdot f(3)$

$\Rightarrow$

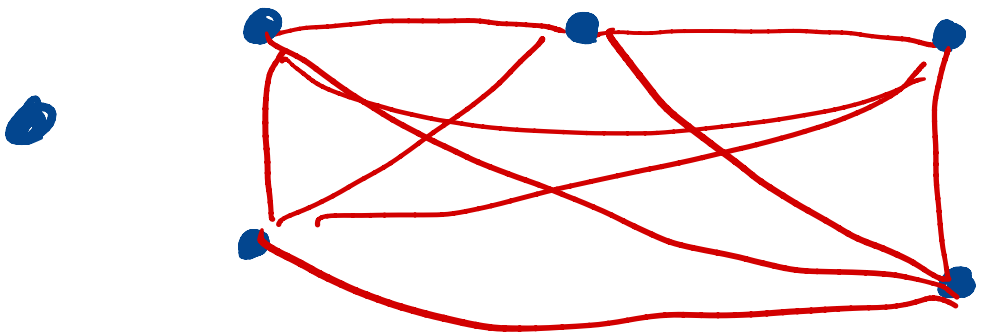
$$f(0) = 1$$
$$f(n) = n f(n-1)$$

Recursion – Handshake example

Example: Suppose n people show up for a meeting. If each person shakes hands with everyone else, how many handshakes occur?

- If there are 2 people in the room, there is 1 handshake.
- If there are 3 people in the room, there are 3 handshakes.
- If there are 4 people in the room, there are 6 handshakes.

Idea: Instead of trying to find a pattern, let's try to define the number of handshakes recursively.



Recursion – Handshake example

Example: Suppose n people show up for a meeting. If each person shakes hands with everyone else, how many handshakes occur?

If there is $n = 1$ person in the room then no handshakes occur.

If there are n people in the room, how many more handshakes are there than there would be if there were $n - 1$ people in the room?

Think of the case when there are $n - 1$ people who arrived on time and shook hands. Then 1 person comes in late. How many people does the new person shake hands with?

Recursion – Handshake example

$H(n)$ Represents the number of handshakes for n people.

Recursive Definition:

$$H(1) = 0$$
$$H(n) = H(n-1) + n - 1$$

Can we come up with an explicit function?

$n = 1: H(1) = 0$
$n = 2: H(2) = H(1) + 1 = 0 + 1 = 1$
$n = 3: H(3) = H(2) + 2 = 1 + 2 = 3$
$n = 4: H(4) = H(3) + 3 = 3 + 3 = 6$
$n = 5: H(5) = H(4) + 4 = 6 + 4 = 10$

$H(n) = H(n-1) + n - 1$

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

$= H(n-3) + (n-3) + n-2 + n-1$

$= H(n-4) + (n-4) + (n-3) + n-2 + n-1$

$= H(n-5) + n-5 + (n-4) + (n-3) + (n-2) + (n-1)$

$\vdots$

$= H(n-(n-1)) + (n-(n-1)) + 2 + 3 + \cdots + (n-2) + (n-1)$

($H(n-(n-1))$: 0)

Recursion – Handshake example

sum of the first $n-1$ integers

$$H(n) = 0 + 1 + 2 + 3 + \cdots + (n-4) + (n-3) + (n-2) + (n-1)$$

Conjecture: If n people, then $H(n) = \frac{n(n-1)}{2}$ handshakes.

Let's prove this with Induction!

Recursion – Handshake example

Base Case: Let $n = 1$, then $H(n) = 0$.

$H(1) = 0$

$H(1) = \frac{1 \cdot (1-1)}{2} = 0$ ✓

for some $k \geq 1$.

Induction Step: Assume that $H(k) = k(k-1)/2$. We want to show that $H(k+1) = k(k+1)/2$ ★

$H(k+1) = H(k) + k$ from the recurrence relation •

$= \frac{k(k-1)}{2} + k$ by our inductive hypothesis

$= \frac{k(k-1)}{2} + \frac{2}{2} \cdot k$

$= \frac{k^2 - k + 2k}{2}$

$= \frac{k^2 + k}{2} = \frac{k(k+1)}{2}$

Therefore, by weak induction we have proven that $H(n) = \frac{n(n-1)}{2}$ for $n \geq 1$

Recursion – Defining Sets

We can also define sets recursively.

Base Step: Start with one element in the set.
Recursive Step: Specify a rule to define more elements in the set.

Example: What set S is described in the following way?
Base Step: $3 \in S$
Recursive Step: $x \in S$ and $y \in S$ then $x + y \in S$

$3 \in S$ Note: $\{3\} = \{3, 3\}$

if $3 \in S$ and $3 \in S$

then $3+3 \in S \Rightarrow 6 \in S$

$\Rightarrow 3+6 = 9 \in S \Rightarrow 3+9 \in S$

This set S describes all positive multiples of 3.

Recursion – Defining Sets

Example: Give a recursive definition of the set E of all even integers.

Base Step: $2 \in E$

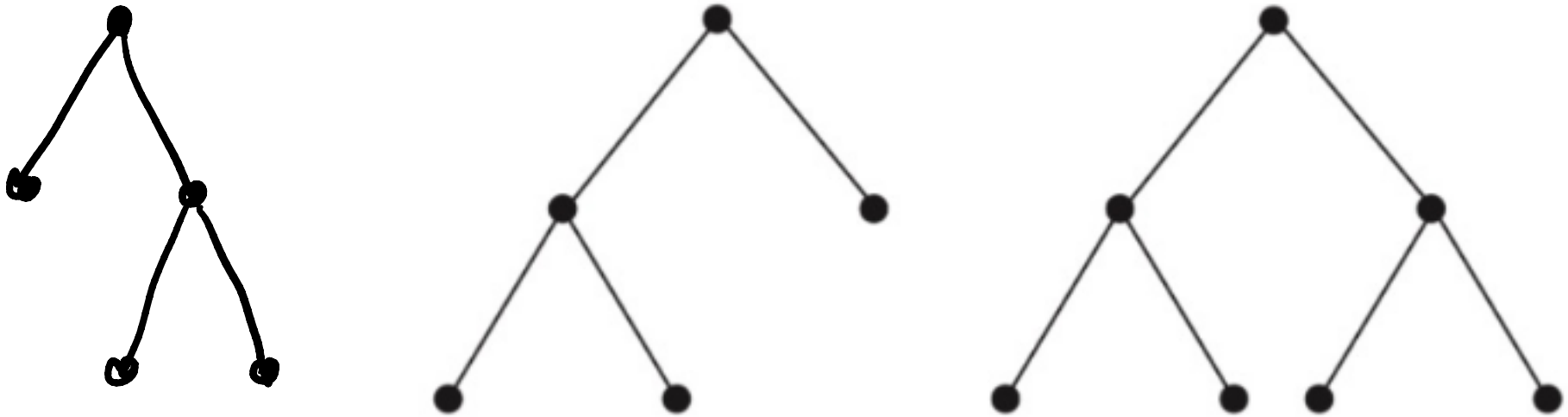
Recursive Step: If $x \in E$, then $x-2 \in E$ and $x+2 \in E$

Recursion – Binary Trees

Binary Trees:

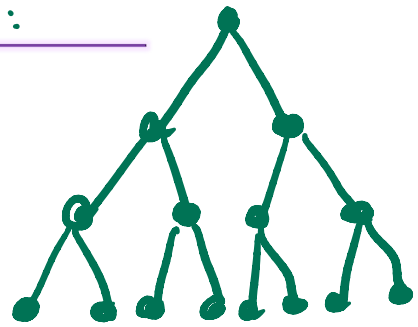
Binary Trees are a type of graph characterized by:

- Begin at a single vertex called a *root*
- Each vertex has at most two children
- We refer to a binary tree as a full binary tree if each vertex has exactly 0 or 2 children



Recursion – Binary Trees

Step 3 :



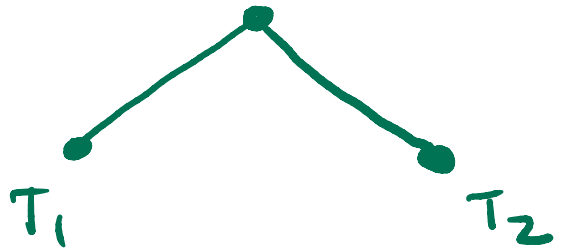
Example: Create the set of all full binary trees by recursion.

Base Step: There is a full binary tree consisting of just a root r

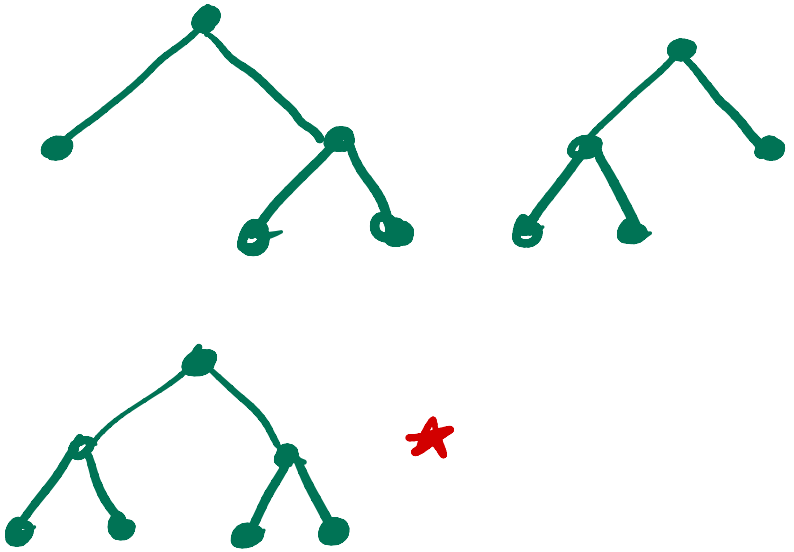
Recursion Step: If T_1 and T_2 are full binary trees, then there exists a full binary tree, denoted $T_1 \cdot T_2$, created by a root vertex connected to the roots of T_1 and T_2.

Base Step:

Step 1 :



Step 2 :



Recursion

Example: Write a recursive algorithm for computing $f(n) = n!$

$f(n) = n!$

$f(0) = 1$

$f(n) = n\,f(n-1)$

One implementation in Python might be:

In [6]:
```
def fact(n):
    if (n==0) or (n==1):
        return 1
    return n * fact(n-1)
```

Recursion – Tower of Hanoi

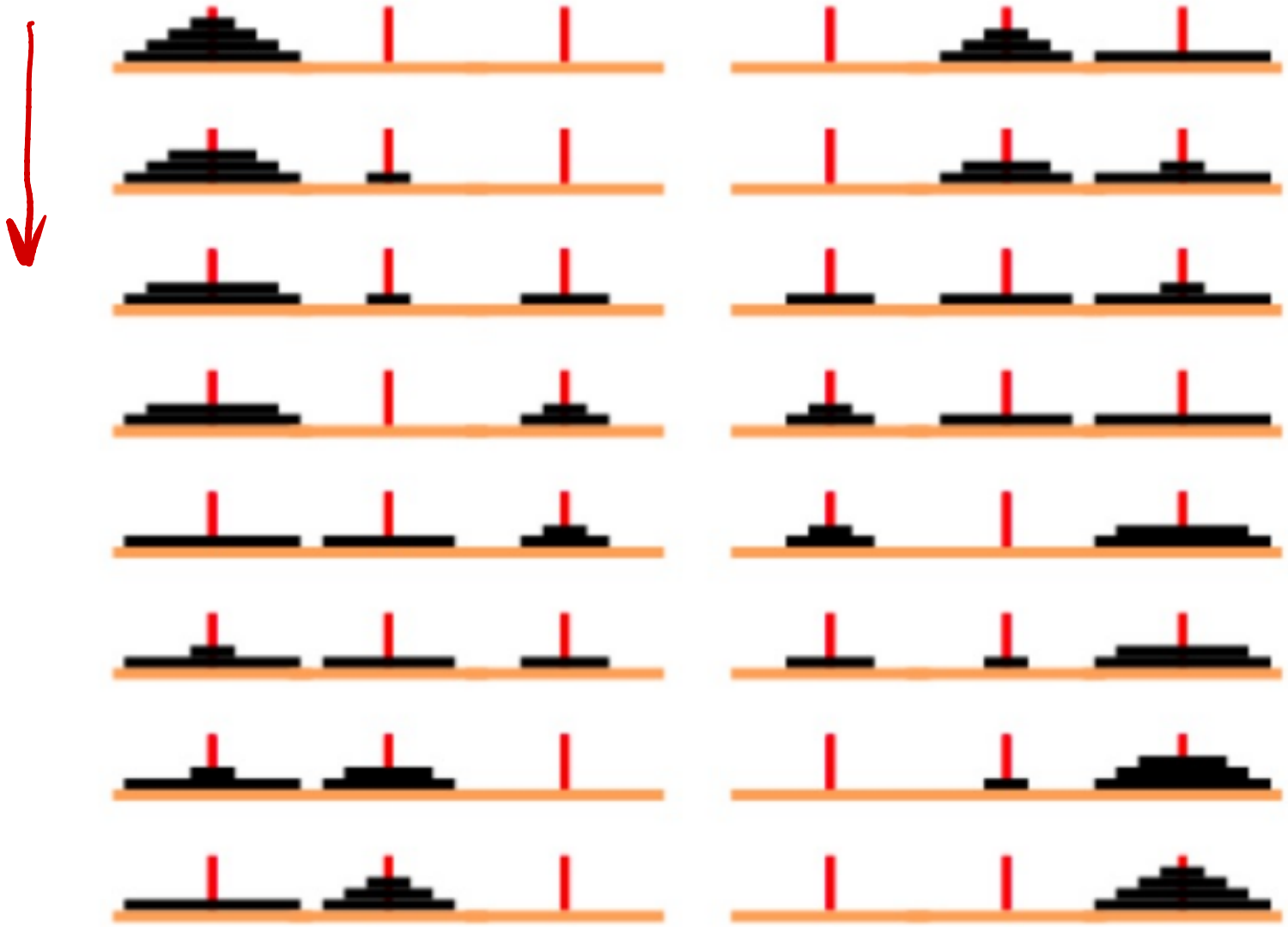
Classic Problem: The Tower of Hanoi

Goal: Move the tower from one peg to another

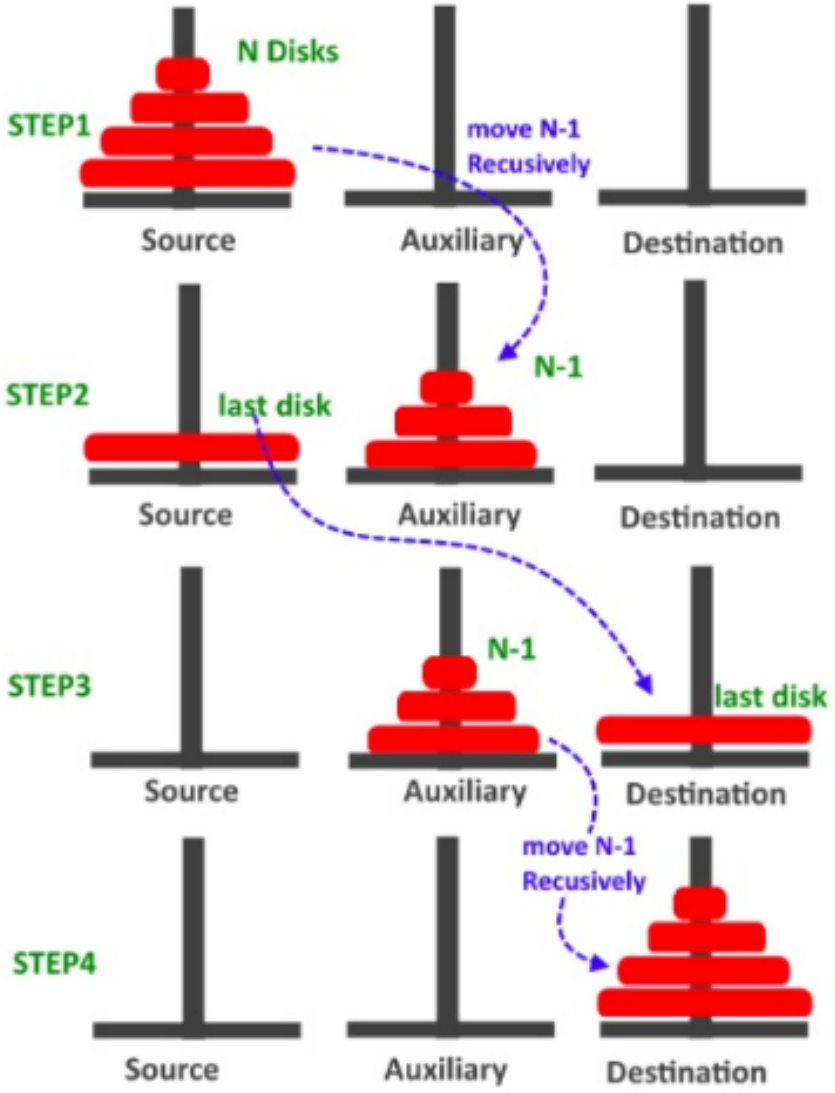
Rules:

- Move one disk at a time
- No larger disk can sit on top smaller disk

Recursion – Tower of Hanoi



Recursion – Tower of Hanoi



Example: Find an algorithm to count the number of distinct moves required to move an n −disk Tower of Hanoi.

- Let $M(n)$ be the number of moves needed to move an *n*-disk Tower.

  Let *n=1*. this takes 1 move.

- To move *n* disks:
  - Move *n-1* disks to the auxiliary peg.
  - Move bottom disk to end peg
  - Move *n-1* disks from auxiliary peg to end peg

Recursion – Tower of Hanoi

$M(1) = 1$

- $M(2) = 2 \cdot M(1) + 1 = 2 \cdot 1 + 1 = 3$
- $M(3) = 2 \cdot M(2) + 1 = 2 \cdot 3 + 1 = 7$
- $M(4) = 2 \cdot M(3) + 1 = 2 \cdot 7 + 1 = 15$

$M(5) = 2 \cdot M(4) + 1 = 2 \cdot 15 + 1 = 31$

$$M(n) = 2\,M(n-1) + 1$$

What's the pattern?

$$M(n) = 2^n - 1$$

Recursion – Tower of Hanoi

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

Prove that $M(n) = 2^n - 1$ by induction.

Base Case: $n = 1$ Recursive def: $M(1) = 1$

$2^1 - 1 = 1$ ✓

Inductive Step: Assume for some $k \geq 1$ that

$$M(k) = 2^k - 1$$

$M(k+1) = 2M(k) + 1$ by recursive def

$= 2(2^k - 1) + 1$ by inductive hyp.