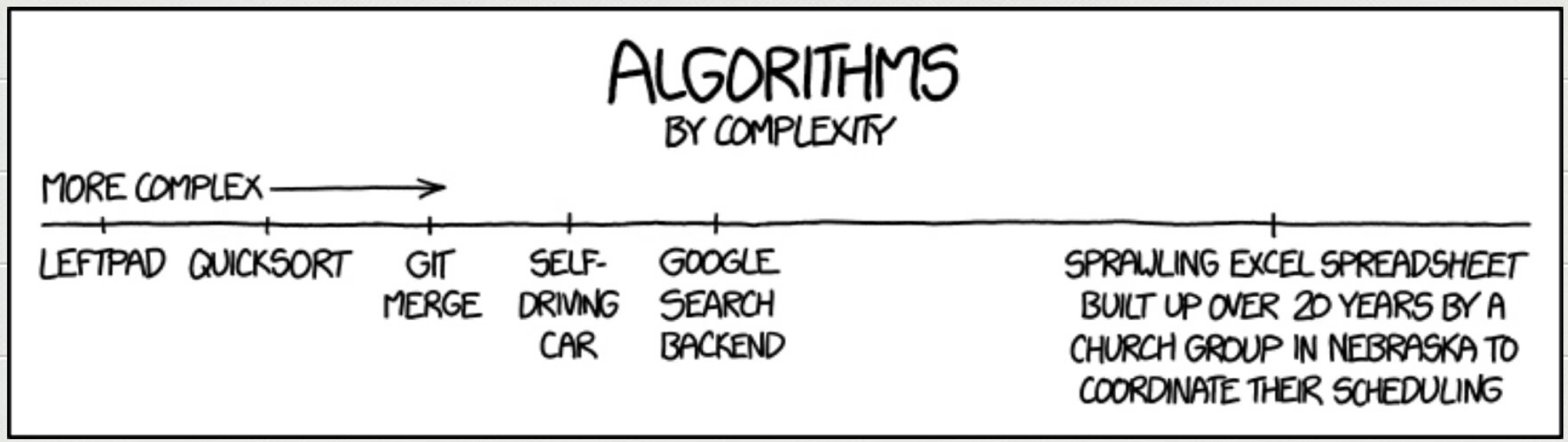
CSCI 3100: Algorithms

Today:

- Big-O
- Algorithm analysis
- Recursion



3 parts to every algorithm:

1.

2.

3.

+ sometimes 4:

This week: why you should have paid attention in discrete math & data structures!

<u>Topics to recall:</u>

Runtimes:

What is big-O analysis?

Why use it?

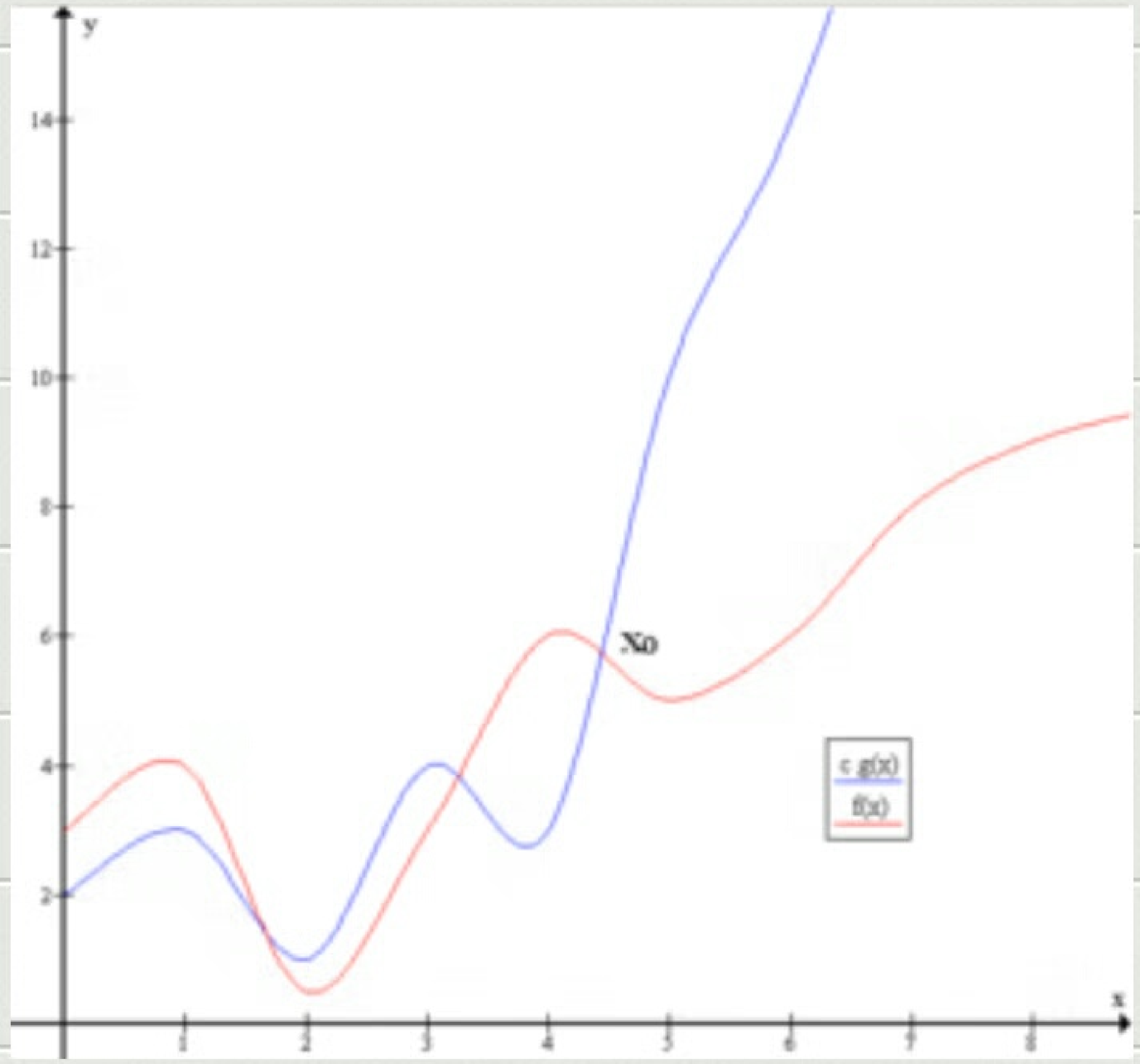
Formal dfn:

Let f & g be functions $\mathbb{R} \to \mathbb{R}$ (or $\mathbb{Z} \to \mathbb{R}$). We say that:

$$f(n) = O(g(n))$$

if $\exists$ constants c & n_0 s.t.

$$|f(n)| \le c|g(n)| \quad \forall n > n_0$$

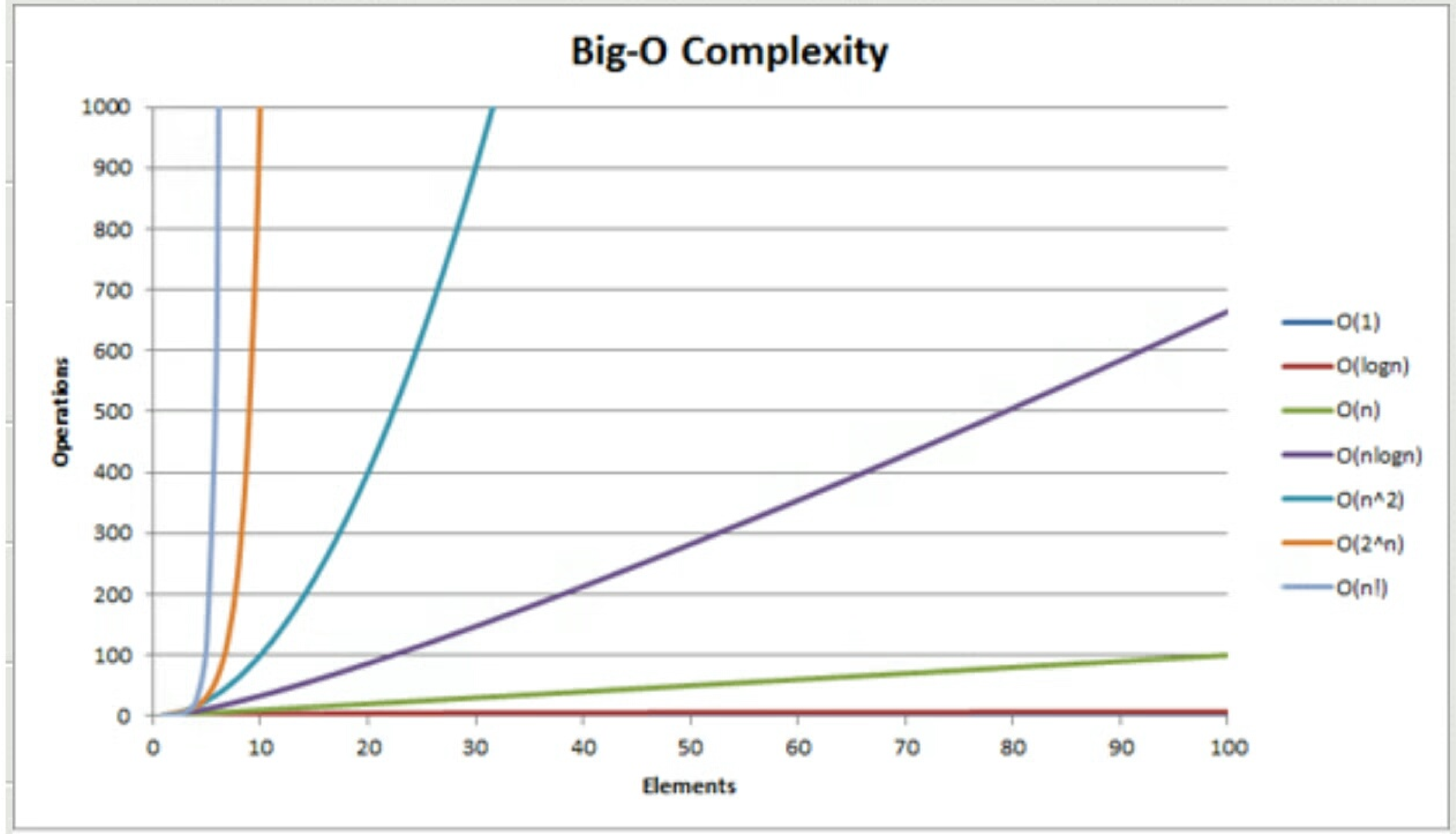


Big-O: functions ranking

BETTER

- $O(1)$ constant time
- $O(\log n)$ log time
- $O(n)$ linear time
- $O(n \log n)$ log linear time
- $O(n^2)$ quadratic time
- $O(n^3)$ cubic time
- $O(2^n)$ exponential time

WORSE



Example proof:

$$f(x) = x^2 + 2x + 1 \text{ is } O(x^2)$$

pf:

Key thm:

Let $f(x)$ be a polynomial of degree n,

so $f(x) = \sum_{i=0}^{n} a_i x^i$

where each $a_i \in \mathbb{R}$.

Then $f(x) = O(x^n)$.

pf sketch:

Induction: recursion's twin

A method of proving a statement which depends on the statement being true for smaller values

Required pieces:

Aside: I think of this as
"automating" a proof:

Show true for $n=1$.

Show if n holds, then $n+1$ must also

$\Rightarrow$ Get all n for free!

Example: $\sum_{i=0}^{n} i =$

Example: The gossip problem

- There are n people, each of whom knows a unique secret
- Every time 2 of them talk, they share every secret they know

Q: How many phone calls are necessary before everyone knows all the secrets?

Thm: For $n \geq 4$, $2n-4$ calls are enough.

Pf:

Now: recursion

- Induction started at the bottom & builds up.

Recursion: the natural dual idea:

Recurrence relations:

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

$$M(n) = 2M\left(\frac{n}{2}\right)+n$$

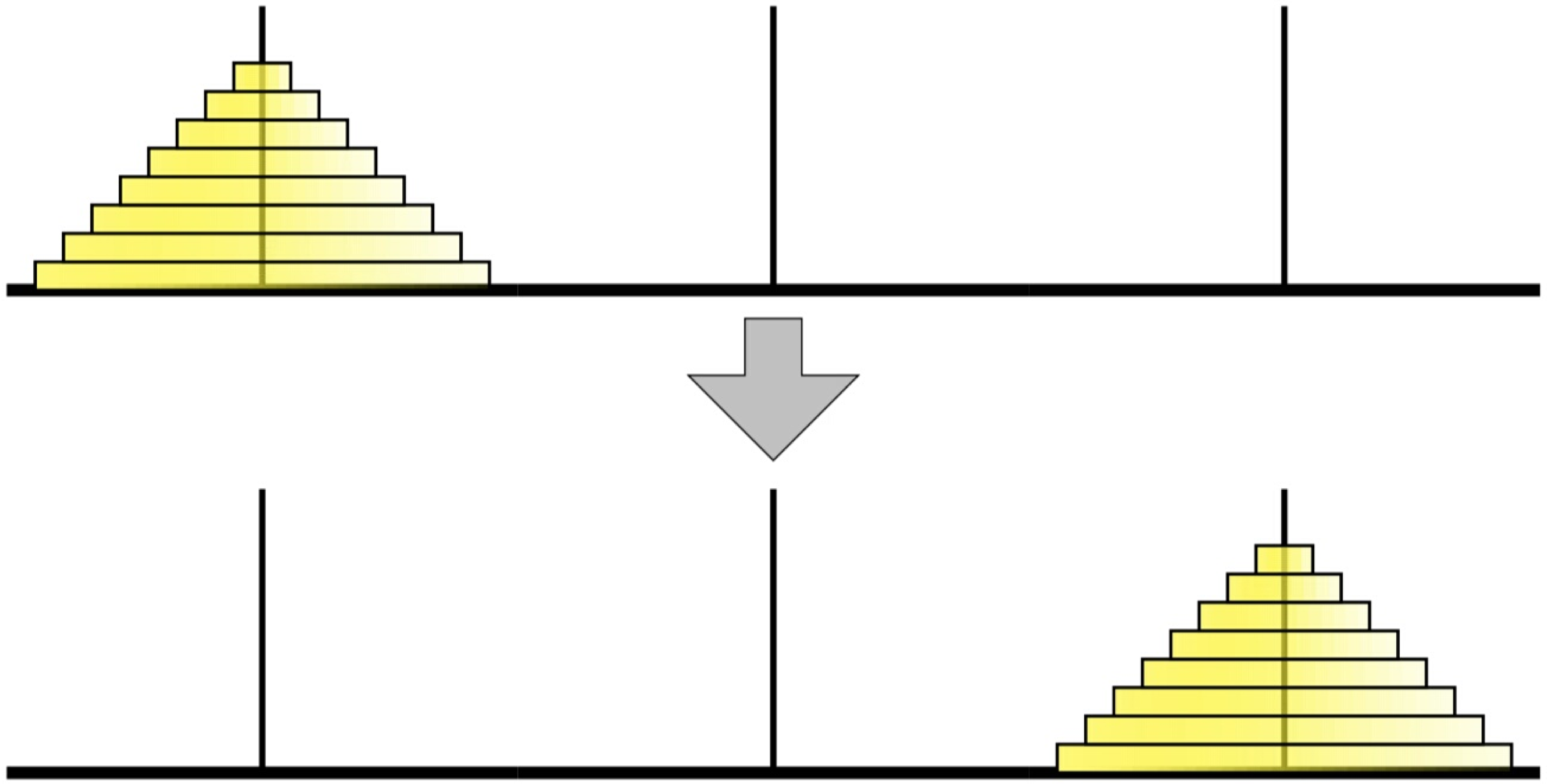
$$T(n) = T\left(\frac{3n}{4}\right)+n$$

How to solve?

Recursive algorithms:
  Based on reduction:
    Reduce to a smaller instance of
    the same problem.

  Necessary pieces (like induction):

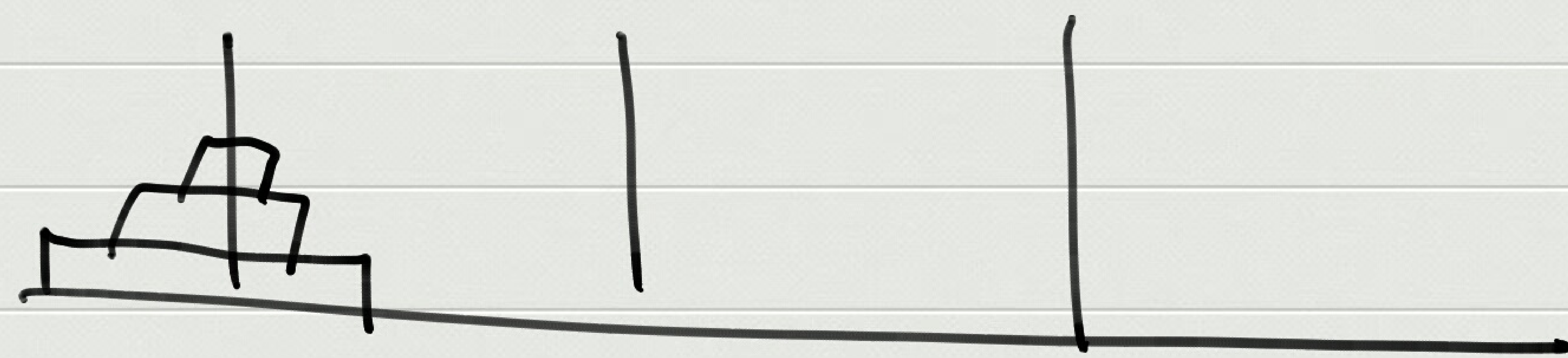
Classical example: Towers of Hanoi



The Tower of Hanoi puzzle

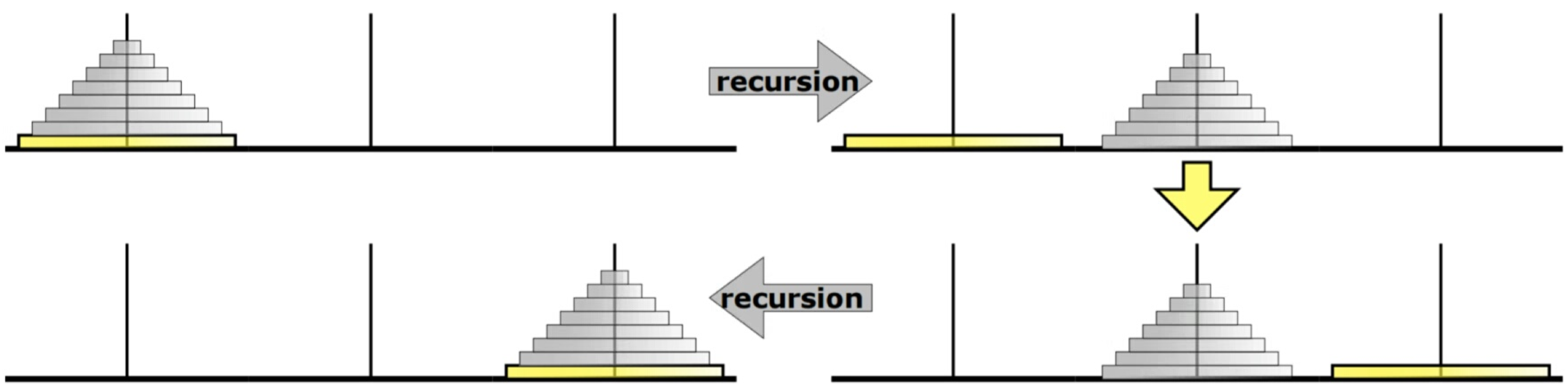
Strategy: think recursively!

Start small:



Bigger picture:

$\underline{\textsc{Hanoi}(n, src, dst, tmp):}$
  if $n > 0$
    $\textsc{Hanoi}(n-1, src, tmp, dst)$
    move disk n from src to dst
    $\textsc{Hanoi}(n-1, tmp, dst, src)$



The Tower of Hanoi algorithm; ignore everything but the bottom disk

Proof of correctness:

Runtime:

Next time:

- Merge sort
- Master thm
- Other classical algorithms