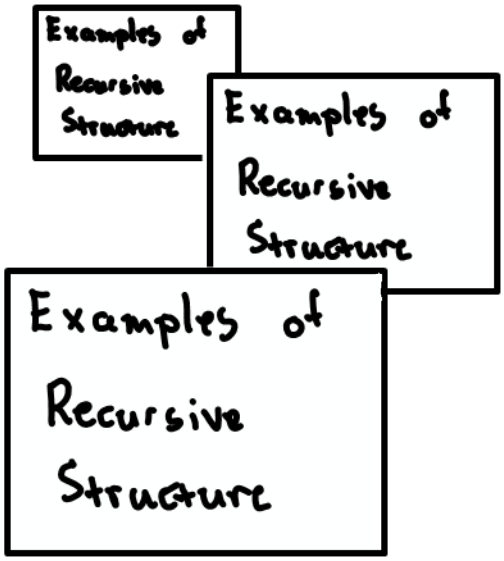
# Discrete Math

## Lecture 20




**example**

$$a_n = 2 \cdot a_{n-1}$$

$$a_0 = 1$$

$1, 2, 4, 8, \ldots$

knowledge of future terms is based on previous terms

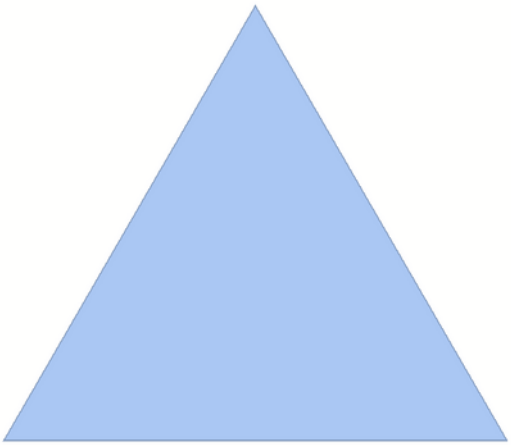
an object where "smaller" or "previous" parts tells us about "bigger" or "future" parts

**ex** the set $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$

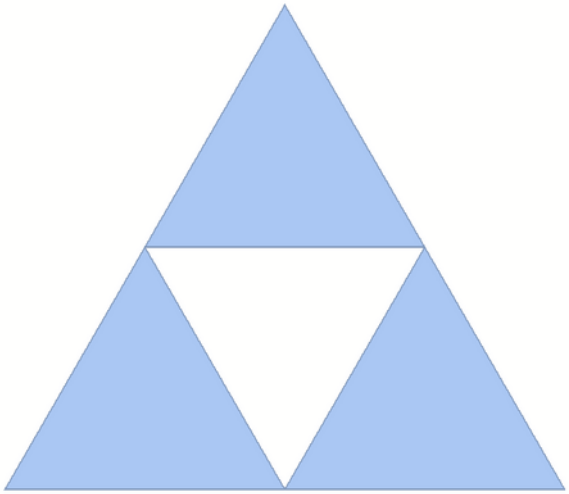
always a "next element" → just add 1

$$\mathbb{N} = \underbrace{\underbrace{\{0\}}_{S_0} \cup \underbrace{\{0+1\}}_{S_1}}_{S_2} \cup \{0+1+1\} \cup \ldots \bigcup_{n=0}^{\infty} S_n$$

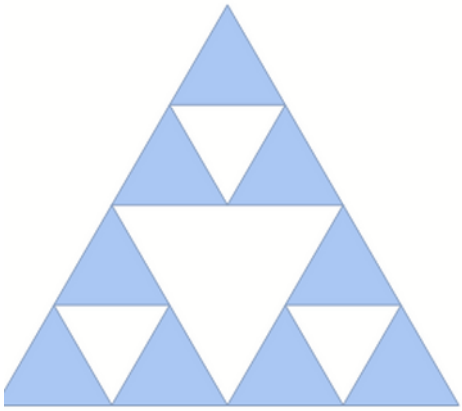
# ex/ Sierpinski's Triangle



Stage 0



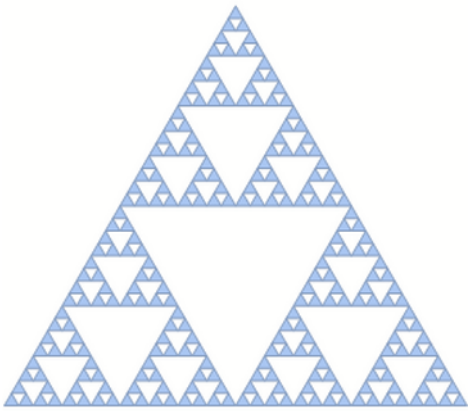
Stage 1



Stage 2

...

more times



iterate a process repeatedly

→ recursive structure

Stage n is related to previous ones

ex] consider the function $f: \mathbb{N} \to \mathbb{N}$

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ 3n+1 & \text{if } n \text{ is odd} \end{cases}$$

(Collatz)

$f(1) = 3 \cdot 1 + 1 = 4$

$f(4) = 4/2 = 2$

$f(2) = 2/2 = 1$

$f(1) = 4$

$1 \to 4 \to 2 \to 1 \to 4 \to 2 \to 1$

$5 \to 16 \to 8 \to \boxed{4 \to 2 \to 1} \to 4 \to 2 \to 1 \ldots$

repeats

No one knows if this always happens!

Spoiler: this is a recursively-defined sequence!

$a_n = f(a_{n-1})$

$a_0 =$ whatever you want to start w/

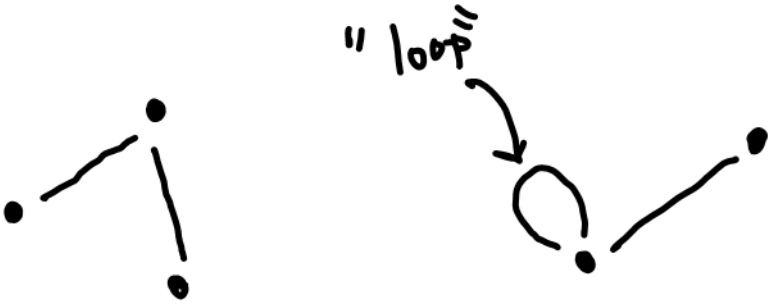
ex) Graph (not a function graph!)

$$G = (V, E)$$

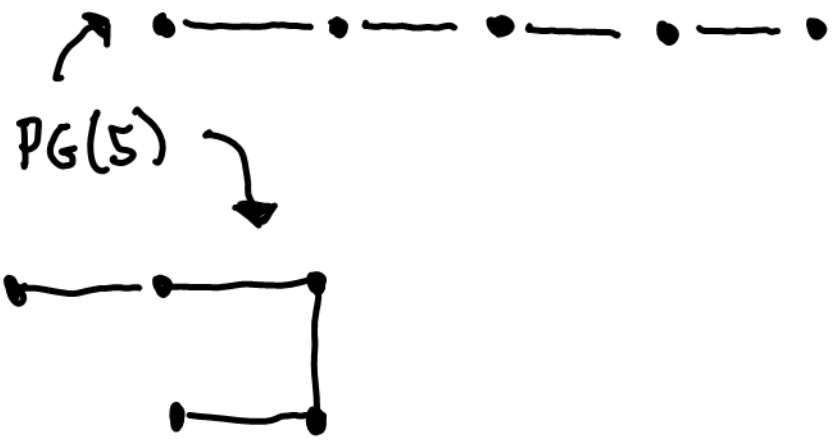
V: vertices or nodes

E: edges

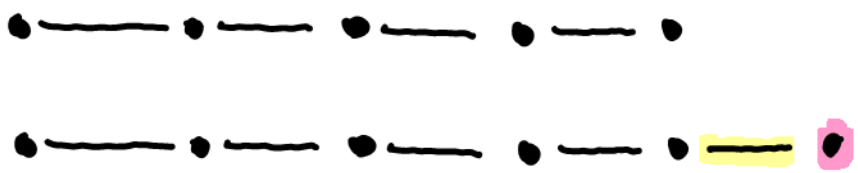
visualize as "dots" connected by lines / curves to other dots




"Path Graph"



PG(6) is related to PG(5)

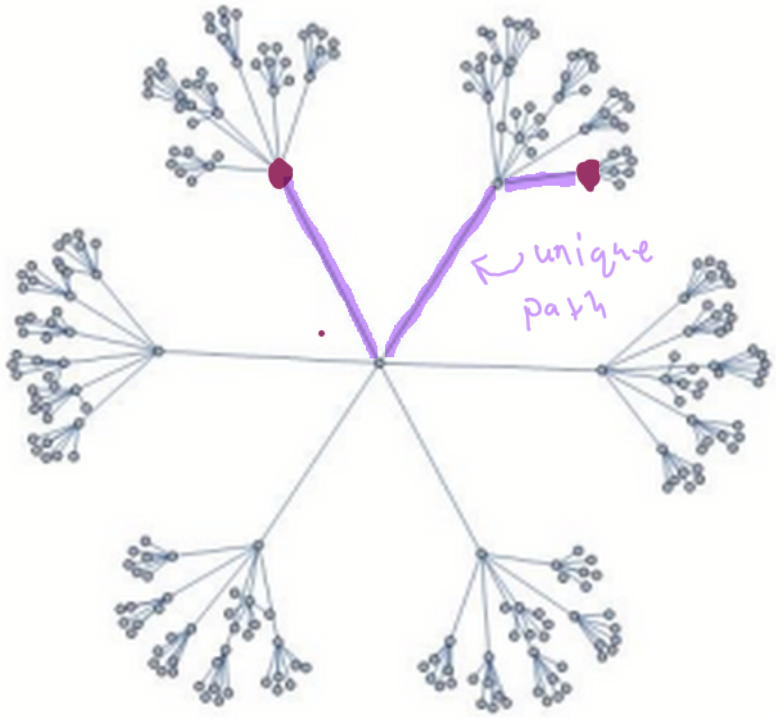


PG(6) = PG(5) + one vertex + one edge

PG(n) = PG(n-1) + one vertex + one edge

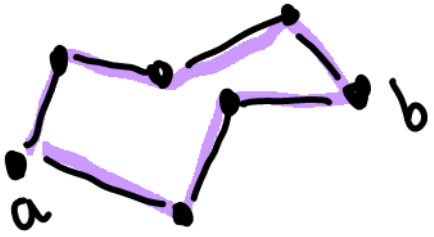
ex/ (rooted, ordered) trees

↳ a graph where every pair of distinct vertices is joined by ! path (seq. of edges)

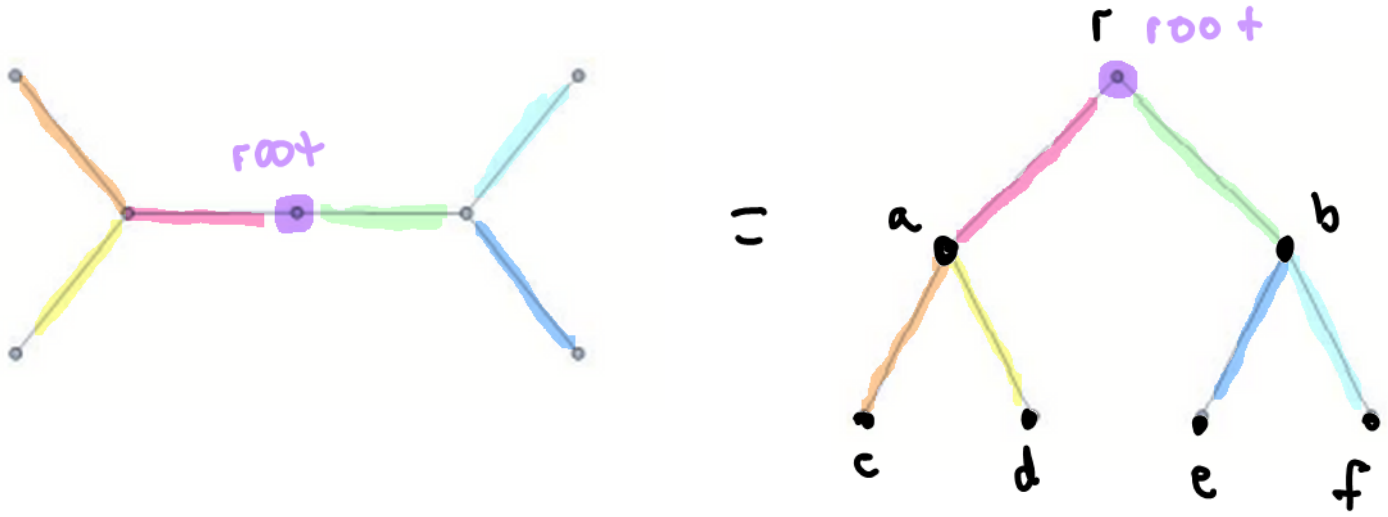



⟺

no "cycles"



a rooted tree has one vertex that's been singled out & called the "root"




a is a "child of" r

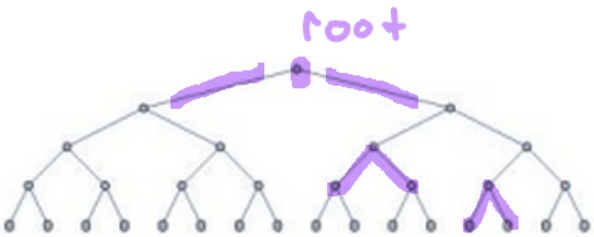
b is a "child of" r

c, d are the children of a

e, f are the children of b

## Full Binary Tree

every node is connected to exactly 2 or 0 nodes




this object is made up of a bunch of identical pieces:

the "whole" is understandable in terms of "parts"

recursive structure!