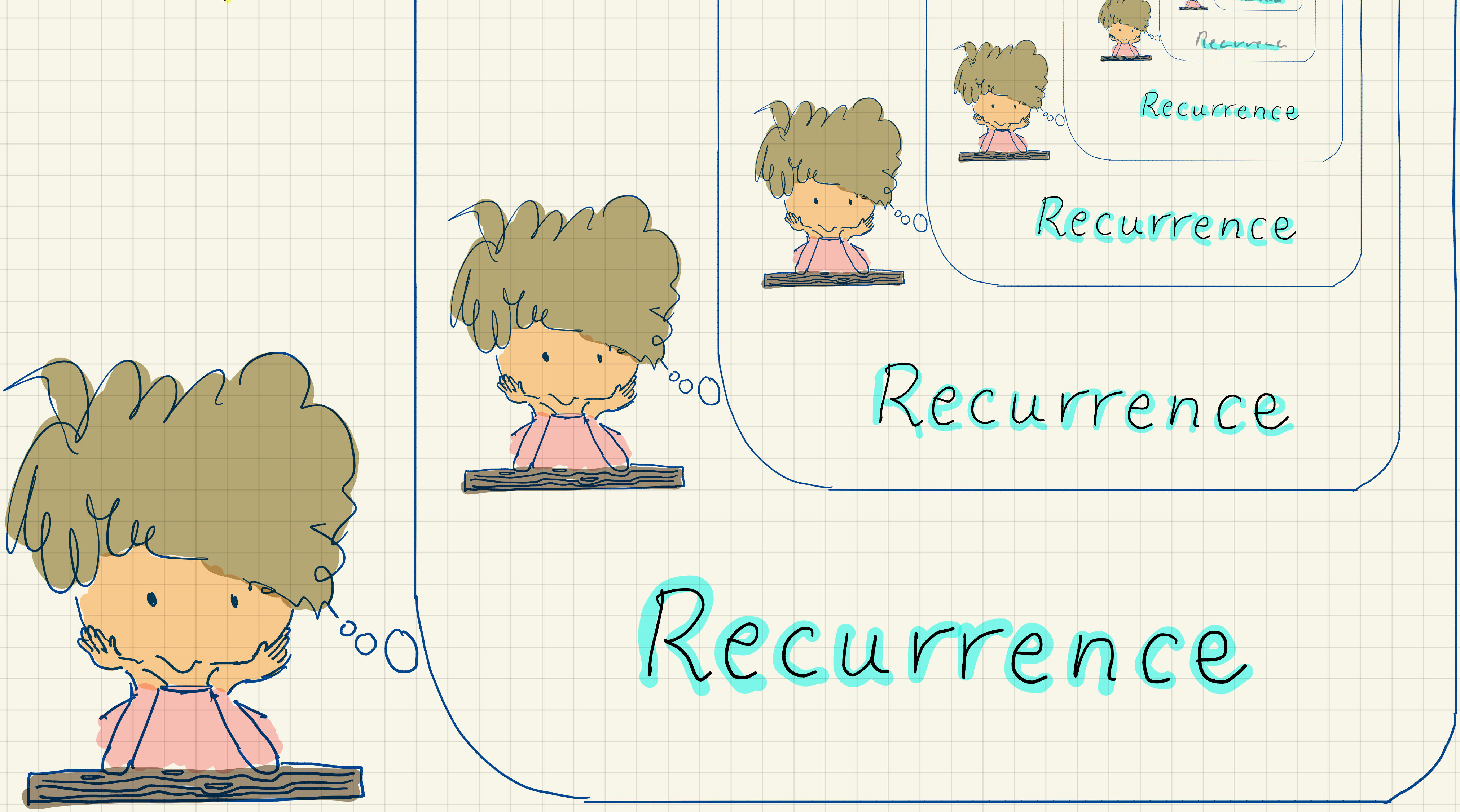
lecture 17



lecture 17
lecture 17
lecture 17
lecture 17
Recurrence
Recurrence
Recurrence
Recurrence

Recurrences & Proofs by Induction

Given an infinite sequence

$$a_0, a_1, a_2, a_3, \ldots$$

where

- the first few values are given $a_0, a_1, \ldots, a_{n_0}$
- a_n is described in terms of $a_{n-1}, a_{n-2}, \ldots$ (a recurrence)

To prove that $a_n = f(n)$ we can use strong induction:

- Base case: verify $a_n = f(n)$ for $n \leq n_0$
- Inductive step: $\forall k \geq n_0 . \bigwedge_{i \leq k} P(i) \implies P(k+1)$

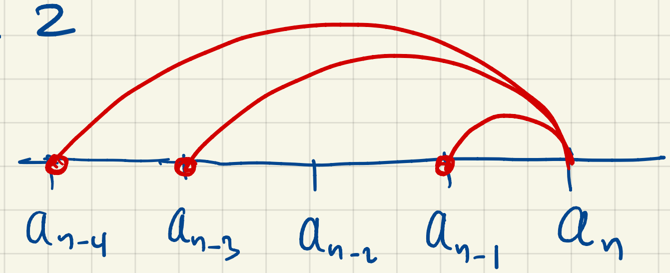
  Consider $a_{k+1} = \ldots$ using the recurrence

  and replace each a_i by $f(i)$

Example:

$$a_0 = 0 \qquad a_1 = 0 \qquad a_2 = 1 \qquad a_3 = 2$$

$$a_n = 2a_{n-1} - 2a_{n-3} + a_{n-4}$$



Prove $a_n = \dfrac{2n^2 - 1 + (-1)^n}{8}$ for $n \geqslant n_0$

Base case:

$$a_0 = \frac{2\times 0 - 1 + (-1)^0}{8} = \frac{0 - 1 + 1}{8} = 0 \checkmark$$

$$a_1 = \frac{2\times 1 - 1 + (-1)^1}{8} = \frac{2 - 1 - 1}{8} = 0 \checkmark$$

$$a_2 = \frac{2\times 4 - 1 + (-1)^2}{8} = \frac{8 - 1 + 1}{8} = 1 \checkmark$$

$$a_3 = \frac{2\times 9 - 1 + (-1)^3}{8} = \frac{18 - 1 - 1}{8} = 2 \checkmark$$

Inductive step: Consider a_{k+1} $\longleftrightarrow$ $\dfrac{2(k+1)^2-1+(-1)^{k+1}}{8}$

$$a_{k+1} = 2a_k - 2a_{k-2} + a_{k-3} \qquad \left(a_{(k+1)} = 2a_{(k+1)-1} - 2a_{(k+1)-3} + a_{(k+1)-4}\right)$$

$$= 2\,\frac{2k^2-1+(-1)^k}{8} - 2\,\frac{2(k-2)^2-1+(-1)^{k-2}}{8} + \frac{2(k-3)^2-1+(-1)^{k-3}}{8}$$

: $4k^2 - 4(k-2)^2 + 2(k-3)^2 = 4k^2 - 4(k^2+4-4k) + 2(k^2+9-6k)$

$$= 4k^2 - 4k^2 - 16 + 16k + 2k^2 + 18 - 12k$$

$$= 2k^2 + 4k + 2 = 2(k+1)^2$$

: $2(-1) - 2(-1) + (-1) = -1$

: $2(-1)^k - 2(-1)^{k-2} + (-1)^{k-3} = -2(-1)^{k+1} + 2(-1)^{k+1} + (-1)^{k+1} = (-1)^{k+1}$

Another example: [simple]

$a_1 = 1$

$a_n = 2a_{n-1} + 1$

Prove $a_n = 2^n - 1$

Base case: $a_1 = 2^1 - 1 = 1$ ✓ $(n_0 = 1)$

Inductive step: $a_{k+1} = 2a_k + 1 = 2[2^k - 1] + 1$

$= 2^{k+1} - 2 + 1 = 2^{k+1} - 1.$

Counting with recurrences

- A recurrence can provide a useful mechanism for counting
- Let's say a_n reprensents some count, as a function of n.
- Sometimes, it's easier to describe a_n by a recurrence
- Guess what a_n is by making enough observations
- Prove it by Induction

We will explore examples of this framework

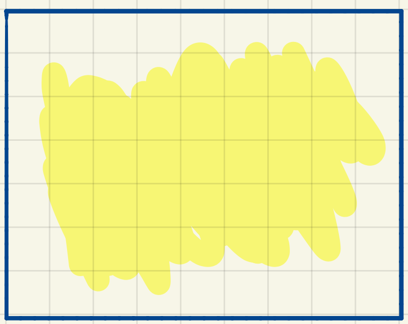
Example 1:

Given n lines in general position

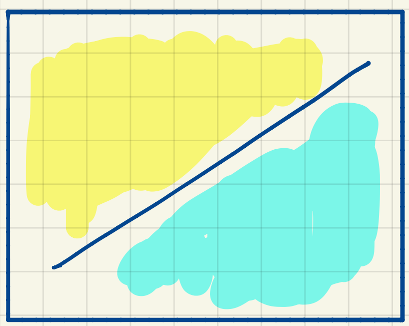
- No lines are parallel
- No 3 lines intersect in a single point.

Basically, every pair of lines have their own intersection.

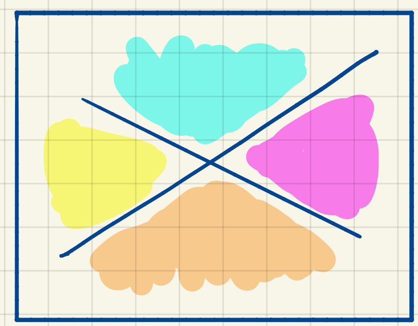
How many regions do the n lines define in the plane?



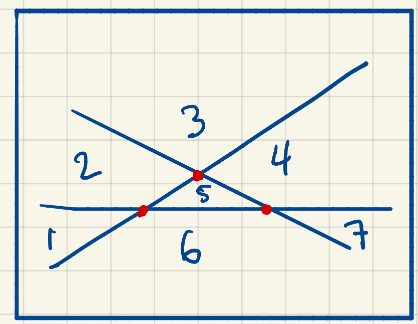
0 lines



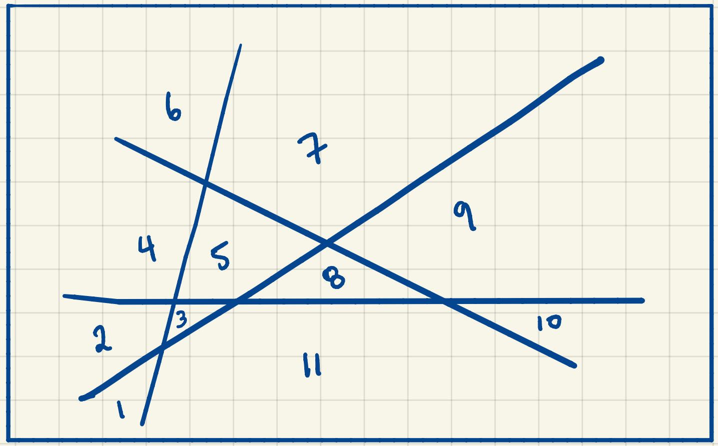
1 line



2 lines



3 lines



4 lines

$R_0 = 1$

$R_1 = 2$

$R_2 = 4$

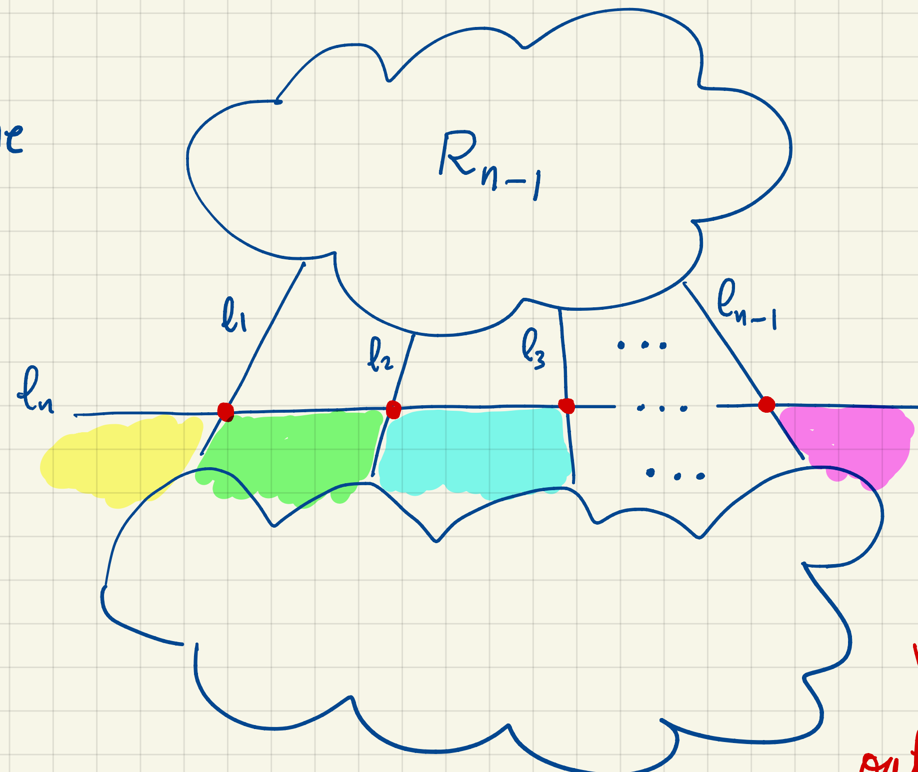
$R_3 = 7$

$R_4 = 11$

Let R_n = # regions defined by n lines

When the n^{th} line is added ...

It creates $n-1$ new intersections and n new regions



It's easier to argue that the n^{th} line adds n new regions than to figure out the exact # regions.

So $R_0 = 1$

$R_n = R_{n-1} + n$

n	0	1	2	3	4	5	6	7	8
R_n	1	2	4	7	11	16	22	29	37

Guess $R_n = 1 + \sum_{i=1}^{n} i = 1 + \frac{n(n+1)}{2}$

Base Case: $R_0 = 1 + \frac{0(0+1)}{2} = 1$ ✓

Inductive Step: $R_{k+1} = R_k + (k+1) = 1 + \frac{k(k+1)}{2} + k+1$

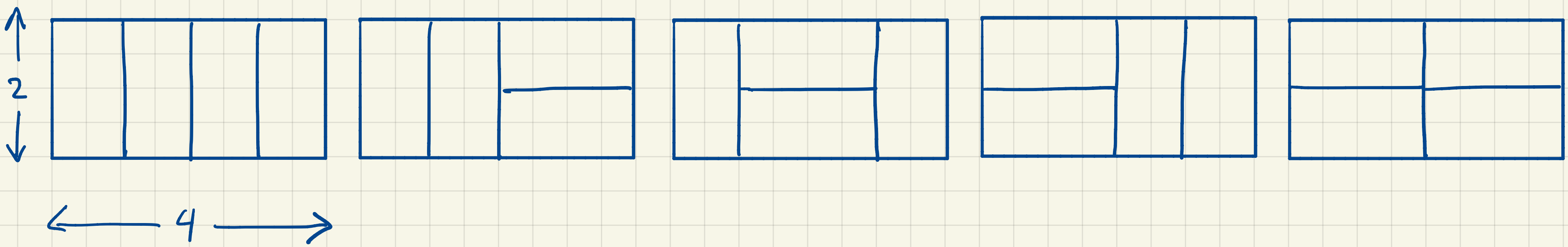
$= 1 + \frac{k(k+1) + 2(k+1)}{2} = 1 + \frac{(k+1)(k+2)}{2}$

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

Example 2. Tiling with Dominos.

In how many ways can I tile a $2 \times n$ rectangle using 2×1 dominos. (we need n dominos)

Let's look at the case of $n = 4$.

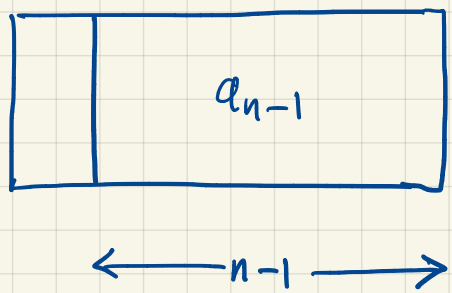


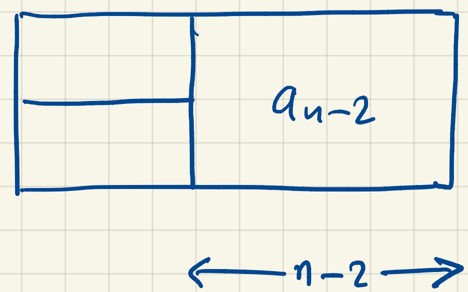
Looks complicated!

But let's figure out a rearrence ...

Let a_n = # ways we can tile a rectangle of length n.

There are two cases, depending on how we start.





In the first case, we continue in a_{n-1} ways

In the second case, we continue in a_{n-2} ways.

Since the cases are disjoint (they start differently), using the addition rule

$$a_n = a_{n-1} + a_{n-2}$$

(Not necessarily Fibonacci)

So $a_1 = 1$

$a_2 = 2$

$a_n = a_{n-1} + a_{n-2}$

n	1	2	3	4	5	6	7
	1	2	3	5	8	13	21

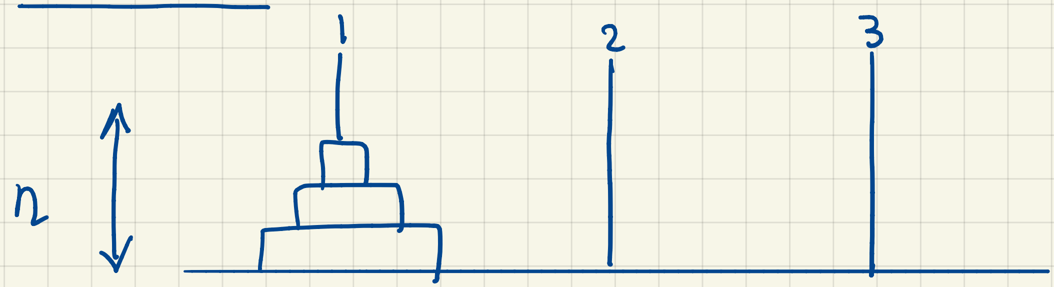
Guess $a_n = F_{n+1}$

Base case : . . .

Inductive step : $a_{k+1} = a_k + a_{k-1} = F_{k+1} + F_k = F_{k+2}$

So $a_{k+1} = F_{(k+1)+1}$.

Example 3 Tower of Hanoi



- Disks numbered 1 through n from smallest to largest
- All disks are stacked on first peg (see above) with disk 1 on top and disk n at bottom.
- Must move entire stack of disks to last peg
  - one disk at a time
  - without ever placing a disk on top of a smaller one.
- How many moves are needed ?

Spoiler Alert.

If $n = 64$ and each move takes 1 second,

we need 585 billion years to finish!

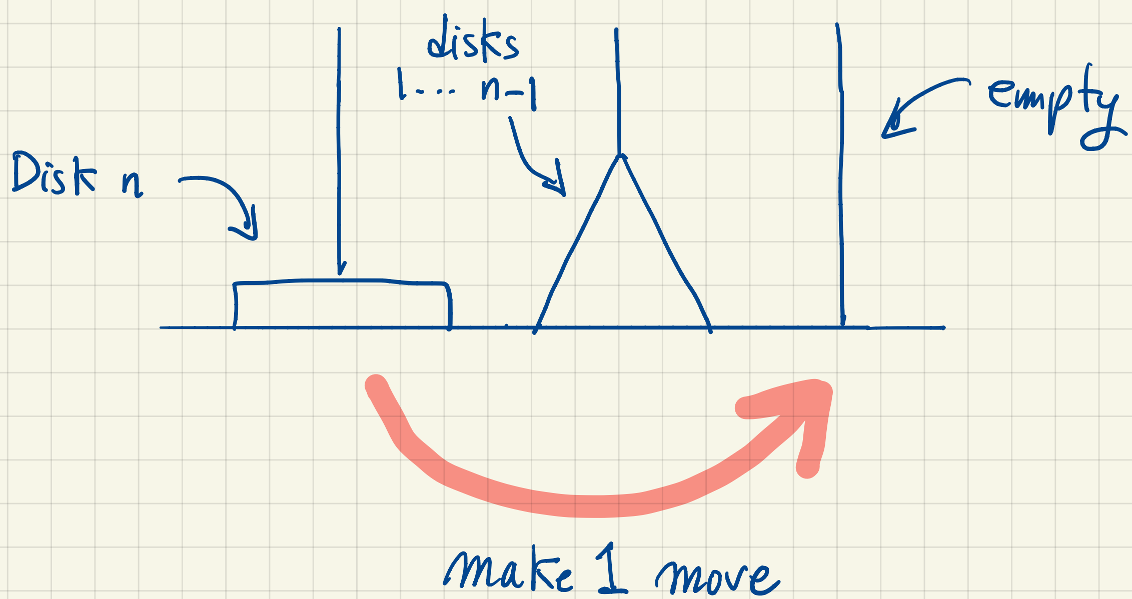
The current estimate of the age of the Universe is

≈ 13.8 billion years.

So this game is not practical to play, and yet we can still study it!

Again, let a_n = # moves needed to solve the puzzle.

The key observation is that at some point, we must move disk n (the largest). So we must have:



- We must move $(n-1)$ disks from first peg to second (Disk n will not be in the way)
  That's a_{n-1} moves by definition

- Then we move disk n (1 move)

- Then we must move $(n-1)$ disks from second peg to third (Disk n will not be in the way)
  That's a_{n-1} moves again.

So $a_n = a_{n-1} + 1 + a_{n-1} = 2a_{n-1} + 1$

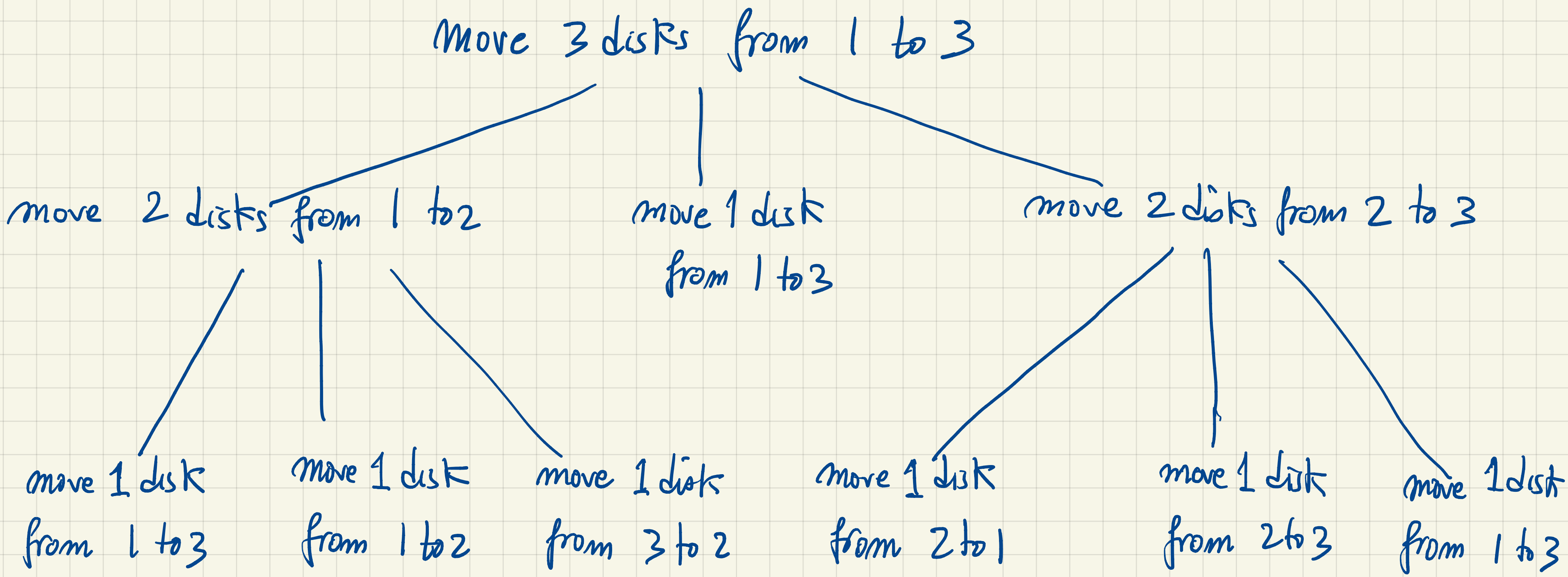
$a_1 = 1$ (trivial)

Solved before : $a_n = 2^n - 1$

- Did we just compute the number of moves or solved the puzzle as well?

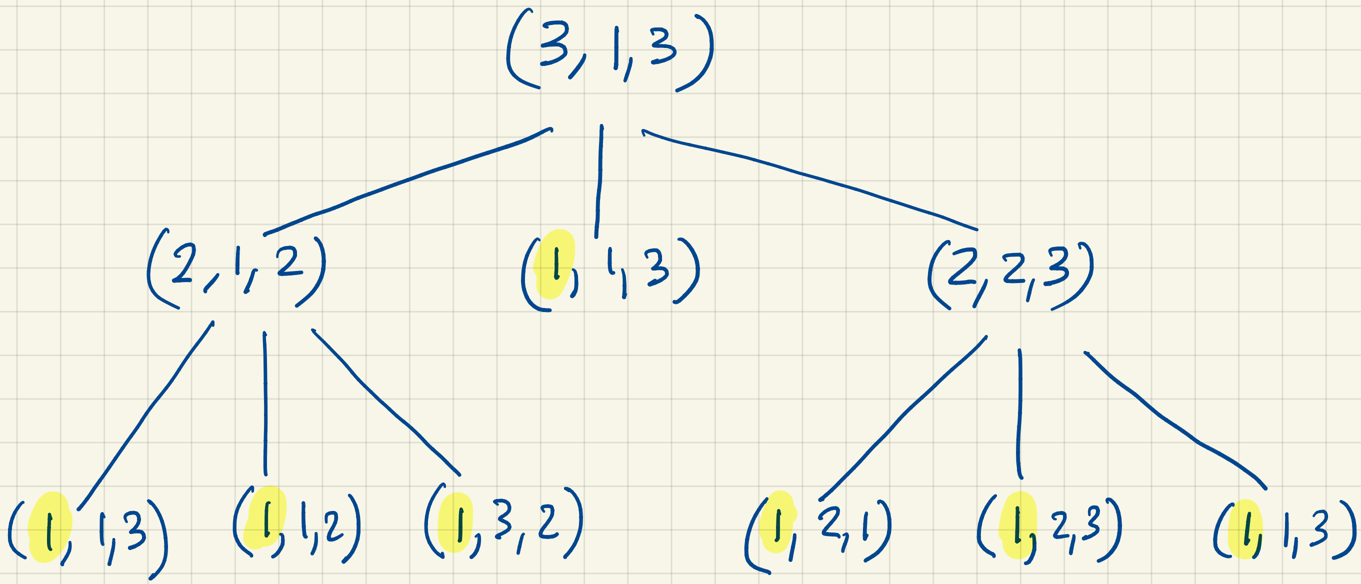
- We actually solved the puzzle, but the solution is described recursively.

- To move n disks from 1 to 3
  - move $n-1$ disks from 1 to 2 (recursive)
  - move 1 disk from 1 to 3
  - move $n-1$ disks from 2 to 3 (recursive)



1 → 3   1 → 2   3 → 2   1 → 3   2 → 1   2 → 3   1 → 3

(# disks, from, to)



1 → 3   1 → 2   3 → 2   1 → 3   2 → 1   2 → 3   1 → 3