16 February 2012

Recurrence

ex. A pair of mature rabbits is placed on an empty island.

Every year: ⇒ all young rabbits grow up

⇒ all mature rabbits produce one young pair

experiment:

year	1	2	3	4	5	6	7
old	1	1	2	3	5	8	13
young	0	1	1	2	3	5	8
total	1	2	3	5	8	13	21

} notice the similarities between the rows (of same elements?)

If we let a_n be the number of pairs of rabbit in year n, then

$a_n = a_{n-1} + a_{n-2}$ (unless $n=2$ or $n=1$)

⟹ "recurrence, recursion"

→ recurrence pattern, however, is not too useful in finding a_{2000}

→ describes interior mechanism ... not the explicit formula

→ we need to know the initial condition that initiates the mechanism:

ex $a_1 = 1$ (similar to differential equation?)

$a_2 = 2$ = express something about the system & its rules.

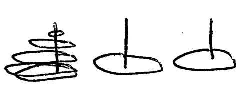
Issues

⇒ Finding a recurrence may be tricky (not all sequences have reasonable occurrence)

⇒ Testing a suggestion for being a solution (easy, but sometimes painful)

⇒ technique for producing an explicit formula from a recurrence

ex Tower of Hanoi

start  end: all washers should be on the last peg...

→ can only move one washer at a time

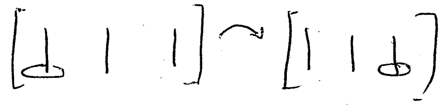
→ you can only put washes on top of bigger washers.

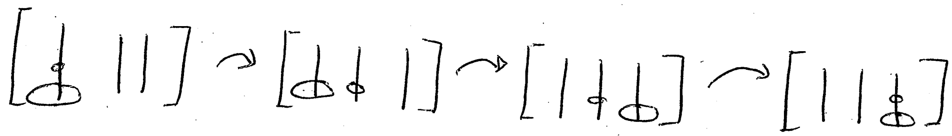
Q: is that possible?

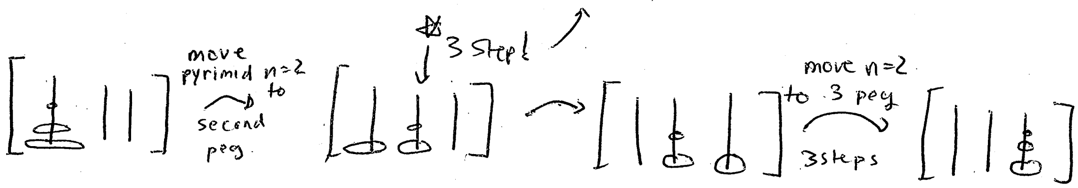
If yes, what is the minimal number of moves?

A: Let a_n = minimum number of moves needed, where n is the number of washers.

We can see:

$n=1$  $= \quad a_1 = 1$

$n=2$  $= \quad a_2 = \cancel{4}\ 3$

$n=3$ 

$a_3 = 3+1+3 = 7.$

$\Rightarrow$ (1) Nothing can be ~~are~~ below the largest
(2) There must be an empty peg
$\Big] \rightarrow$ must move pyrimid of $n=2$ to peg 2.

In general

$$a_n = \underset{\substack{\uparrow \\ 2\times \text{ move pyrmid } n-1}}{2a_{n-1}} + \underset{\substack{\uparrow \\ \text{move large disk}}}{1}$$

] recurrence explains the intricacies of the game (it was not obvious! we needed to be clever)

<u>Def</u> A <u>recurrence</u> is

a formula $a_n = F(a_{n-1}, a_{n-2}, \ldots)$ plus a sufficient number of initial conditions

If $F(-)$ takes an input from the harvest of $a_{n-1}, \ldots, a_{n-d}$ only, we say $a_n = F(a_{n-1}, \ldots, a_{n-d})$ and one with the <u>recurrence depth of d</u>

A recurrence that is truly of depth d needs d initial conditions.

Back to Hanoi

$a_n = 2a_{n-1} + 1 \quad (a_{n-1} = 2a_{n-2} + 1)$

$= 2(2a_{n-2}) + 2 \quad (a_{n-2} = 2a_{n-3} + 1)$

$= 2(2(2a_{n-2}) + 3 + 4$

$\vdots$

$= 2^k a_{n-k} + (1+2+4+\ldots) = 2^k a_{n-k} + (1+2+4+\ldots + 2^{k-1})$

what is this theory?

$\ldots + \frac{1}{8}2^k + \frac{9}{4}2^k +$ half of $2^k = 2^k$

$$= 2^{n-1} a_1 + 2^{n-1} - ?$$

$$= 2 \cdot (2^{n-1}) - 1 = 2^{n-1} \checkmark$$

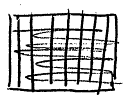
* Note

the above formula was easily found for substitution of the previous term was rather simply, and the resulting pattern was clearly identifiable.

⇒ Not a good technique, in general.

Challenge Qustions

(1) Given is a bar of chocolate



→ you are allowed to break along the cracks but only one piece at a time

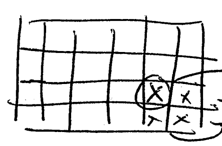
→ best strategy for eating fast.

(2) Given is a bar of chocolate

→ 2 people: player A & B. Taking turn, the player chooses one piece + and eats the piece and the other pieces that are not above it and left of it.

ex

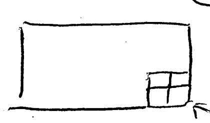
Player A :



Goal? DO NOT EAT THE LAST PIECE

Q: What is the winning strategy?

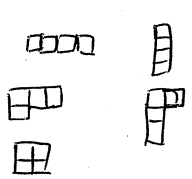
B



(3) Pick a $n \in \mathbb{N}$

Now consider all L-like shapes (an array of n squares all inside the lower right quadrant ( ) and such that each row starts at the y-axis, each row is at least as long as every row below it, + top row along the x-axis

If $n = 4$



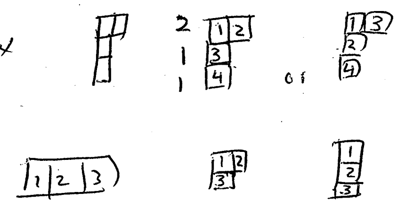
← L-like shapes

= then, the number of squares in each row form a partition of n (non-increasing collection of numbers that add up to n

We now place numbers $1, —, n$ in such a way that from numbers goes up

$\Rightarrow$ left $\rightarrow$ right
top $\rightarrow$ bottom

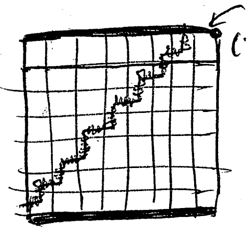
ex



Q: How many such L like shapes w/ contents are there?

$\Rightarrow$ recursion? explicit formula?

Ex A to B in grid : w/ River

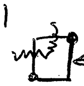


river: cannot be crossed.

= How many efficient ways ?

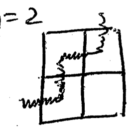
ex 1

$n=1$



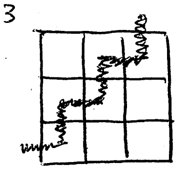
1 way

$n=2$



= 2 ways.

$n=3$

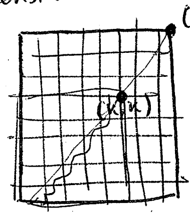


= 5 ways

Finding Recursion:

Start $(0,0)$ and (n,n) are on the diagonal. Classify each walk according to the first time we cross the diagonal again.

Consider



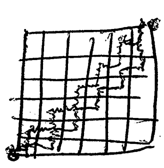
(k,k) = first visit to the diagonal

from (k,k) to (n,n) there are a_{k-n} ways to arrive at (n,n)

How about $(0,0) \rightarrow (k,k)$?

Because we agreed that k,k is our first time of reaching the diagonal, it is as if we have another river flowing through d

(This does not make sense



(To be cont.)