Lecture 12

Recursion and induction
(on numbers… and on structures!)

Recursion and induction

Not a chapter in Schaum.
"Full induction" discussed as proof technique.

Discussions on recursively defined functions in Chapter 3.6.

Recursion is a computation / programming technique, see also *Programming methods (Programmeermethoden) and Algorithmics (Algoritmiek).*

Here we use recursion / induction to define functions / sets of objects.
There is a small dictation available for reading, made by H.J. Hoogeboom (in Dutch). See the website of Fundamentele Informatica 1 of Prof. de Graaf.

We do not deal with the section on co-graphs here.
We also do not address the semantics of arithmetic expressions.

Recursion versus induction

- **Induction:** a way of defining objects and sets of objects — defining bigger object via their smaller parts.
  - We then talk about "inductive definitions".
- **Recursion:** Functions can also be defined using "smaller versions of themselves" (their function on "smaller instances").
  - Then we talk about "recursively defined functions".
- **Induction in proofs:** a way of proving properties of objects which are inductively defined.
- If we use the recursive / inductive definition as a calculation or programming technique to calculate, for example, a recursively defined function, we speak of recursion. Usually has a simple **iterative step.**

Very abstract, will become clear through examples….

Small caveat: Recursion versus induction

Inductive definition: start with the smallest objects and indicate how smaller objects make up larger ones: *from small to large*

Recursive definition: indicate how a larger value / object is made from smaller ones: *from large to small.*

The difference between induction and recursion is therefore often a matter of perspective (and even context).

Recursion and induction

Example; recursively (inductively) defined sequence:

$$a_0 = 0$$
$$a_n = a_{n-1} + n \ (n \geq 1)$$

$$a_0 = 0$$
$$a_1 = a_0 + 1 = 1$$
$$a_2 = a_1 + 2 = 1 + 2 = 3$$
$$a_3 = a_2 + 3 = 3 + 3 = 6$$

iterative….

$0,1,3,6,10\ldots$

“Well-defined”

- The definition of a function is recursive it it (the definition) refers to the function itself in the definition.
- This sounds problematic (circular arguments).
- For the definition to *make sense* … for us to be able *to compute it* (for the function to be “well-defined”) two conditions must be met:
  - We need one or more **basis cases,** where the value of the function is given explicitly (or via other well defined functions).
  - The recursive step **refers to smaller cases of the function** (and other independent objects), so the basic case is eventually reached

$a_0 = 0$	basis
$a_n = a_{n-1} + n \ , \ (n \geq 1)$	recursive (inductive) step

Well-defined

Sorites paradox

what a heap makes…

Inductive, recursive and iterative

$a_0 = 0$
$a_n = a_{n-1} + n \ (n \geq 1)$

Inductive view, "build up"

$$a_0 = 0$$
$$a_1 = a_0 + 1 = 1$$
$$a_2 = a_1 + 2 = 1 + 2 = 3$$
$$a_3 = a_2 + 3 = 3 + 3 = 6$$

Recursive view, reduce, reduce, reduce

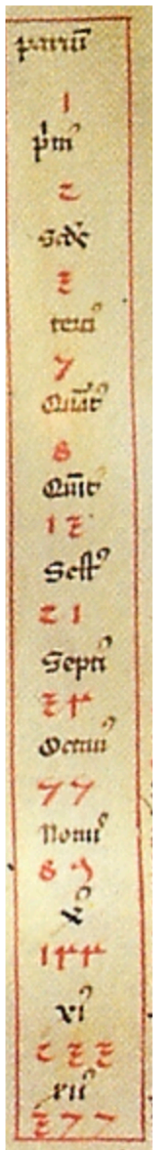
$$a_3 = a_2 + 3$$
$$a_2 = a_1 + 2$$
$$a_1 = a_0 + 1$$
$$a_0 = 0$$

Iterative view…

0,1,3,6,10…

Liber abaci (1202)

historic book on arithmetic by Leonardo of Pisa, son of Bonacci

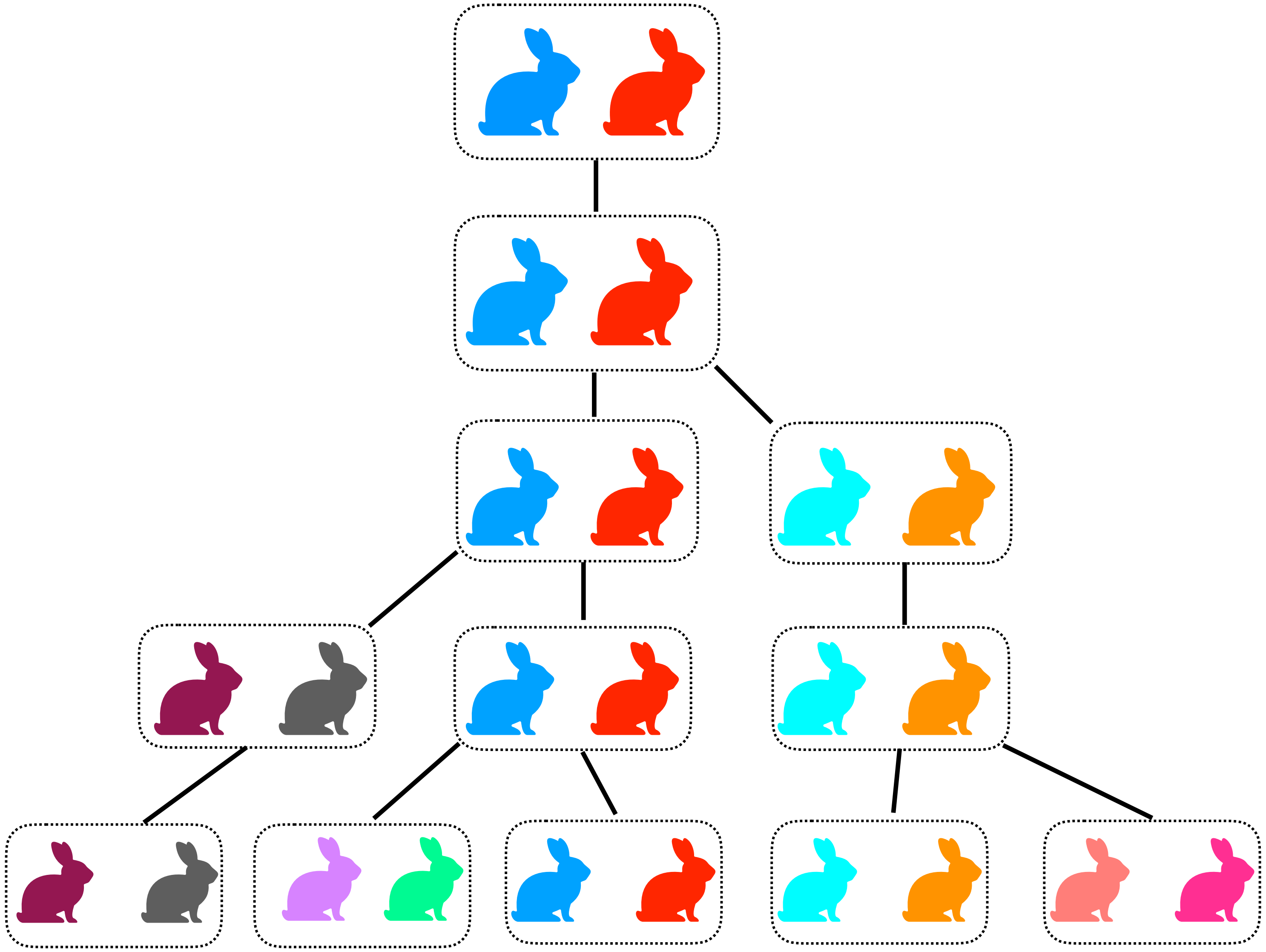


A certain man put a pair of rabbits in a place surrounded on all sides by a wall.

How many pairs of rabbits can be produced from that pair in a year if it is supposed that every month each pair begets a new pair which from the second month on becomes productive?

https://en.wikipedia.org/wiki/Liber_Abaci

From rabbits to deep math..



number of pairs

1

1

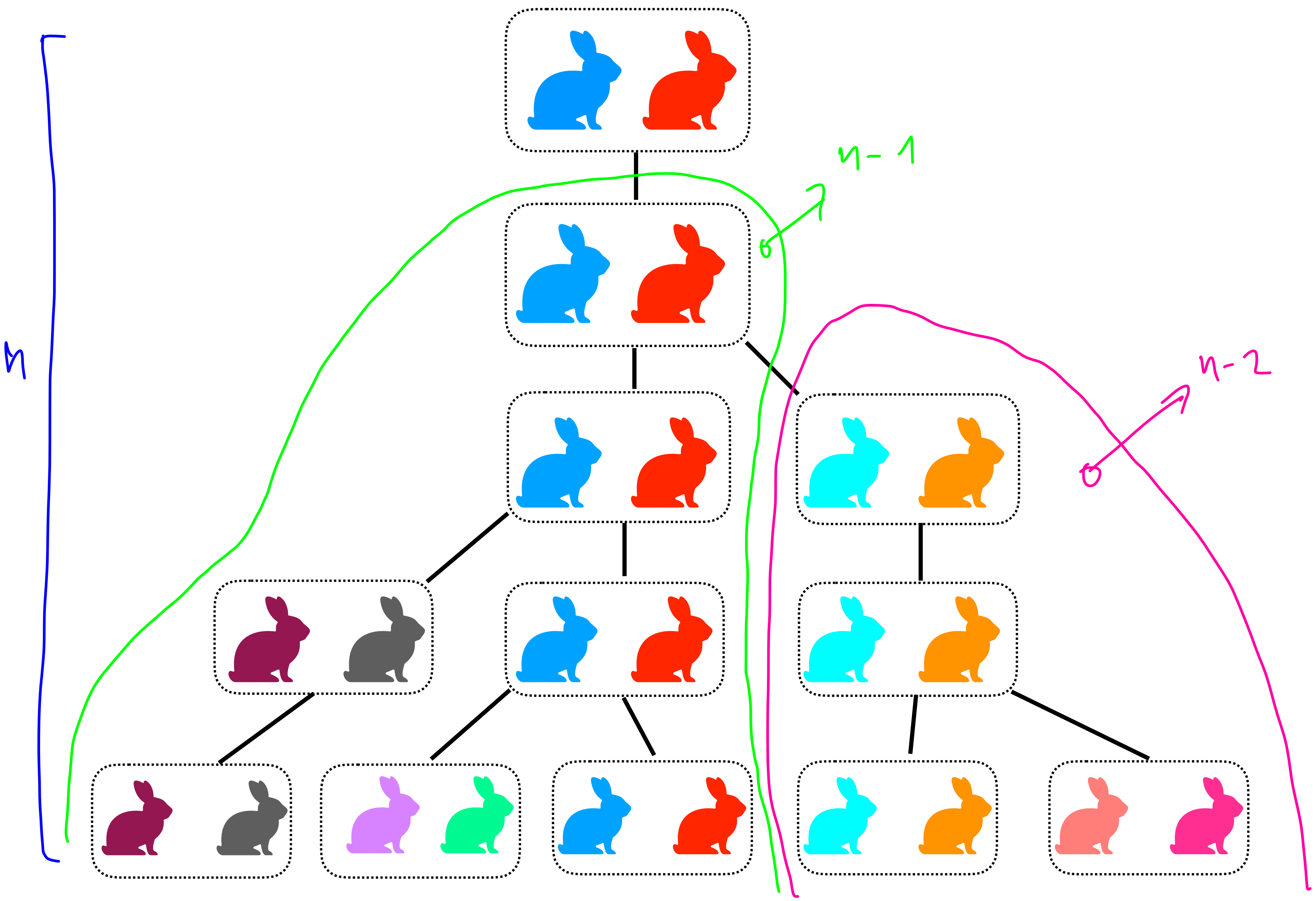
2

3

5

https://en.wikipedia.org/wiki/Liber_Abaci

From rabbits to deep math..



$F_n = F_{n-1} + F_{n-2}$

https://en.wikipedia.org/wiki/Liber_Abaci

Fibonacci numbers

$$F_0 = 0$$
$$F_1 = 1$$
$$F_n = F_{n-1} + F_{n-2}\ ,\ (n > 1)$$

Working F_4 inductively

$F_0 = 0$

$F_1 = 1$

$F_2 = F_1 + F_0 = 1$

$F_3 = F_2 + F_1 = 1 + 1 = 2$

$F_4 = F_3 + F_2 = 2 + 1 = 3$

0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,2584,4181…

0 1 2 3 4

Fibonacci numbers

$$F_0 = 0$$
$$F_1 = 1$$
$$F_n = F_{n-1} + F_{n-2}\ ,\ (n > 1)$$

Working out F_4 recursively

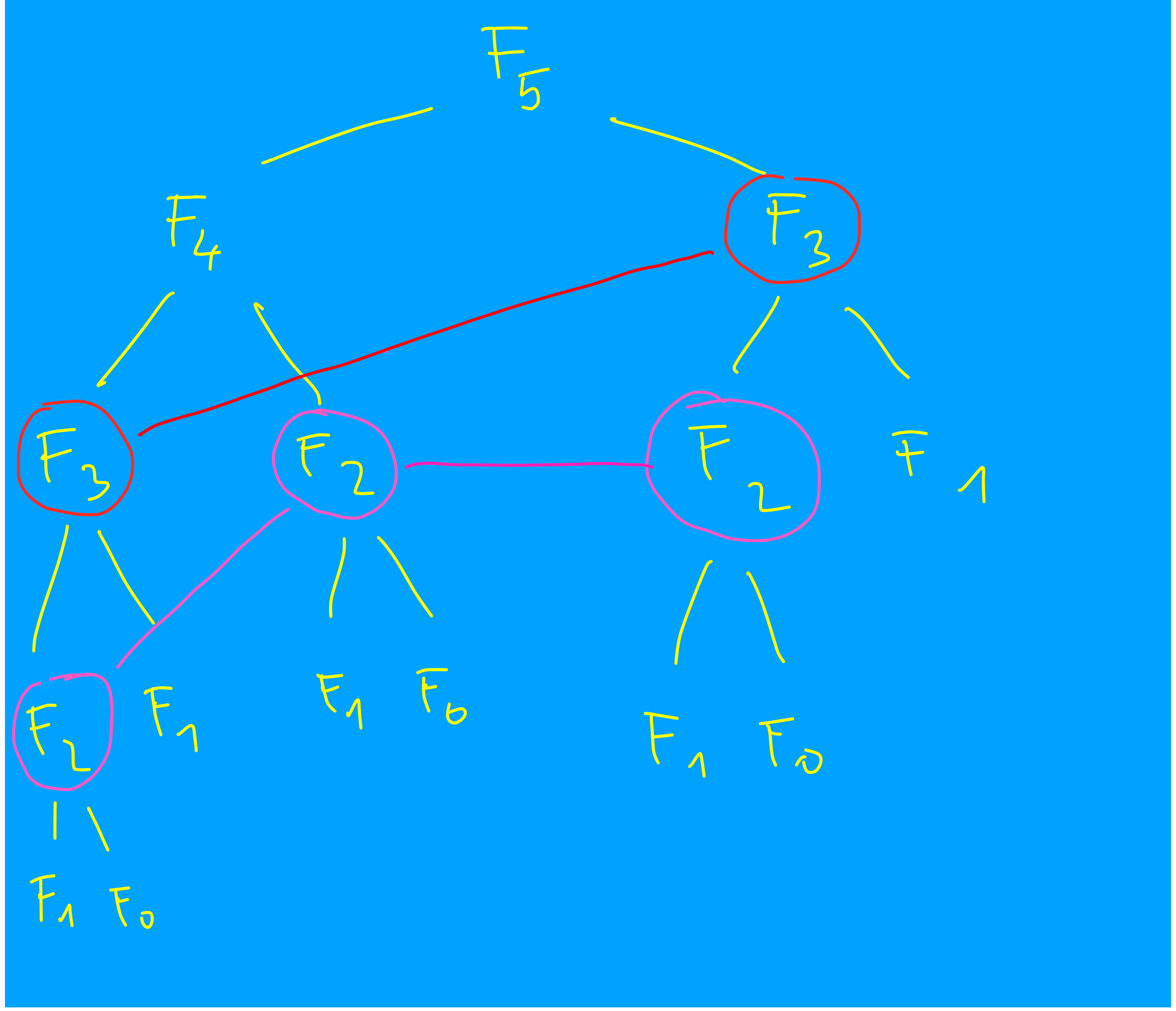
$$F_4 = F_3 + F_2 = F_2 + F_1 + F_1 + F_0 = F_1 + F_0 + F_1 + F_1 + F_0 = 3$$

0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,2584,4181…

Fibonacci numbers

$$F_0 = 0$$
$$F_1 = 1$$
$$F_n = F_{n-1} + F_{n-2}\ ,\ (n > 1)$$

Recursive tree



- Fibonacci numbers are **inductively defined:** we compute **the next from two previous**

- But they can be **computed recursively**. For any desired *n,* express it in terms of previous elements, and repeat for each. If we are not careful we will compute many values many times over…

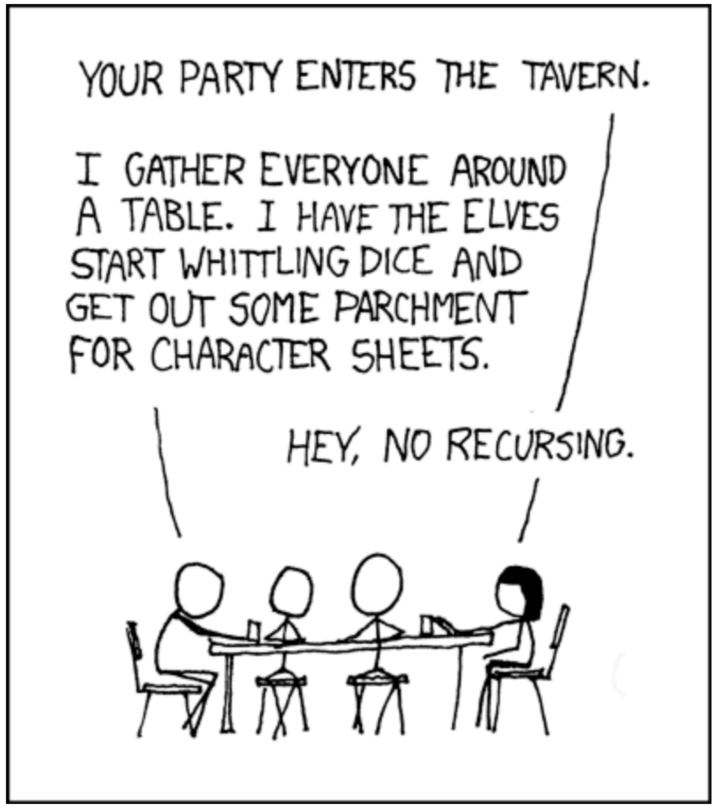
Recursively defined functions

The definition refers to the function itself.

Again, must be well defined:

Usable if the following two requirements must be met:

- one or more basic cases specified directly
- the self-referential part refers to a smaller input value



https://xkcd.com/244/

Recursively defined functions

- $f(0) = 1$
- $f(n) = n \times f(n-1),\ n>0$

Recursively defined functions

- $f(0) = 1$
- $f(n) = n \times f(n-1),\ n>0$

- $f(6) = 6 \times f(5) = 6 \times 5 \times f(4) = \ldots = 6\times5\times4\times3\times2\times1\times f(0) = 6!$

- ***Closed-form expressions, non-recursive definitions***

- $f(n) = n!$
- $f(n) = \prod_{k=1}^{n} k$

Recursively defined functions

- $g(0) = 0$
- $g(n) = n + g(n-1),\ n>0$

Recursively defined functions

- $g(0) = 0$
- $g(n) = n + g(n-1), n>0$

- $g(6) = 6 + g(5) = 6 + 5 + g(4) = \ldots = 6+4+5+3+2+1+0 = 21$

- $g(n) = \sum_{k=1}^{n} k$
- $g(n) = n(n+1)/2$

Recursively defined functions

- *g(0) = 0*
- *g(n) = n + g(n-1), n>0*

  - *g(6) = 6 + g(5) =6 +5 + g(4)= ... =6+4+5+3+2+1+0 = 21*
  - $g(n) = \sum_{k=1}^{n} k$
  - *g(n) = n(n+1)/2*

Going back to Fibonacci

- $F_0 = 0$
- $F_1 = 1$
- $F_{n+1} = F_n + F_{n-1},\ n \geq 1$

- *0, 1, 1, 2, 3, 5, 8, 13, 21, 34 …*

Fibonnacci closed formula?

Going back to Fibonacci

- $F_0 = 0$
- $F_1 = 1$
- $F_{n+1} = F_n + F_{n-1},\ n \geq 1$

- *0, 1, 1, 2, 3, 5, 8, 13, 21, 34 …*

Fibonnacci closed formula?

Binet's formula

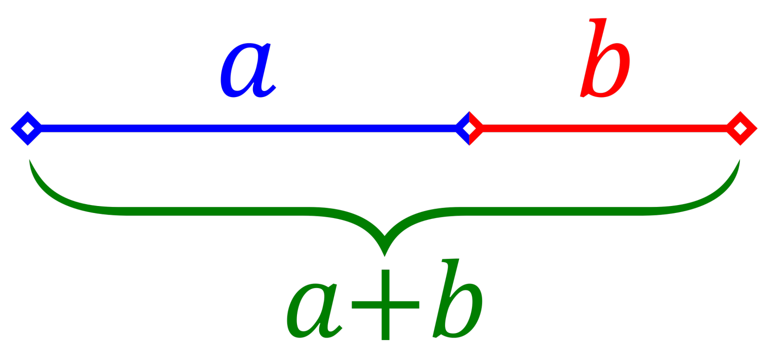
$$F_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$$

Isn't this weird?

Golden ratio

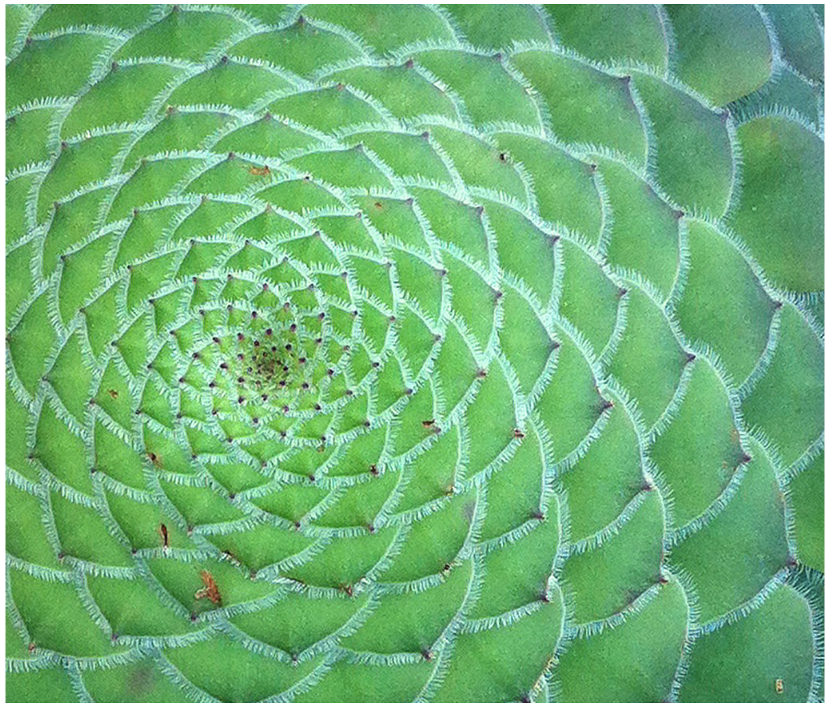
- $x^2 = x + 1$

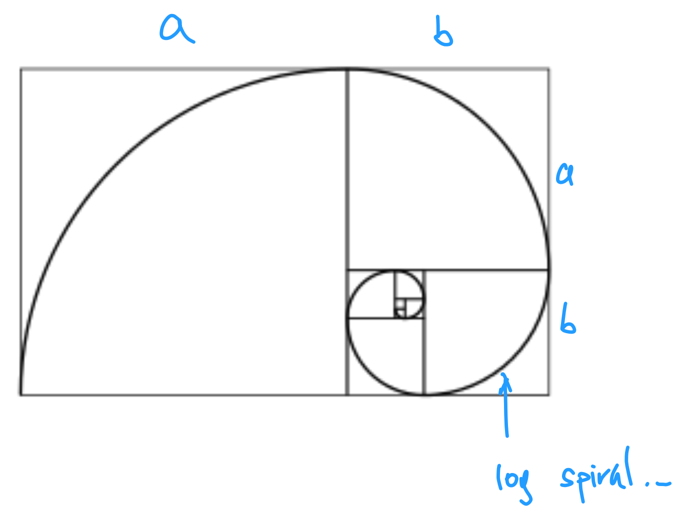
set $b = 1$ …



$a+b$ is to a as a is to b

Fibonacci numbers





Bull Math Biol. 2008 Apr;70(3):643-53. Epub 2007 Nov 10.

Nucleotide frequencies in human genome and fibonacci numbers.

Yamagishi ME[1], Shimabukuro AI.

Author information

Abstract

This work presents a mathematical model that establishes an interesting connection between nucleotide frequencies in human single-stranded DNA and the famous Fibonacci's numbers. The model relies on two assumptions. First, Chargaff's second parity rule should be valid, and second, the nucleotide frequencies should approach limit values when the number of bases is sufficiently large. Under these two hypotheses, it is possible to predict the human nucleotide frequencies with accuracy. This result may be used as evidence to the Fibonacci string model that was proposed to the sequence growth of DNA repetitive sequences. It is noteworthy that the predicted values are solutions of an optimization problem, which is commonplace in many of nature's phenomena.

PMID: 17994268 DOI: 10.1007/s11538-007-9261-6

TradingView.

Recurrence relation — closed expression

- $t_0 = 5,\ t_1 = 6$
- $t_n = t_{n-1} + 6t_{n-2} - 12n + 8, \quad n \geq 2$

Work out the first elements…

Recurrence relation — closed expression

- $t_0 = 5,\ t_1 = 6$
- $t_n = t_{n-1} + 6t_{n-2} - 12n + 8, \quad n \geq 2$

Work out the first elements…

5,
6,
6+30-24+8 = 20,
20+36-36+8 = 28,
28+120-48+8 = 108,
108+168-60+8 = 208,
208+648-72 +8 = 792
…

secret rule?

Recurrence relation — closed expression

- $t_0 = 5,\ t_1 = 6$
- $t_n = t_{n-1} + 6t_{n-2} - 12n + 8, \quad n \geq 2$

Work out the first elements…

5,
6,
6+30-24+8 = 20,
20+36-36+8 = 28,
28+120-48+8 = 108,
108+168-60+8 = 208,
208+648-72 +8 = 792
…

secret rule?

MAYBE ITS;

$t_n = 3^n + (-2)^n + 2n + 3$ ***closed expression***

??

Recurrence relation

- $t_0 = 5,\ t_1 = 6$
- $t_n = t_{n-1} + 6t_{n-2} - 12n + 8, \quad n \geq 2$

Work out the first elements…

Simple recurrence relations allow a (simple) method for solving…
For this example, the secret recipe is in Schaum, Section 6.8,
by solving 2nd order homogeneous linear recurrence relations.
The above recurrence relation is not homogeneous,
but the secret recipe can be extended to non-homogeneous second order recurrence relation.

THIS IS NOT GOING INTO THE EXAM

$$t_n = 3^n + (-2)^n + 2n + 3$$ ***closed expression***

IS IT TRUE? DEBT.

Recurrence relation

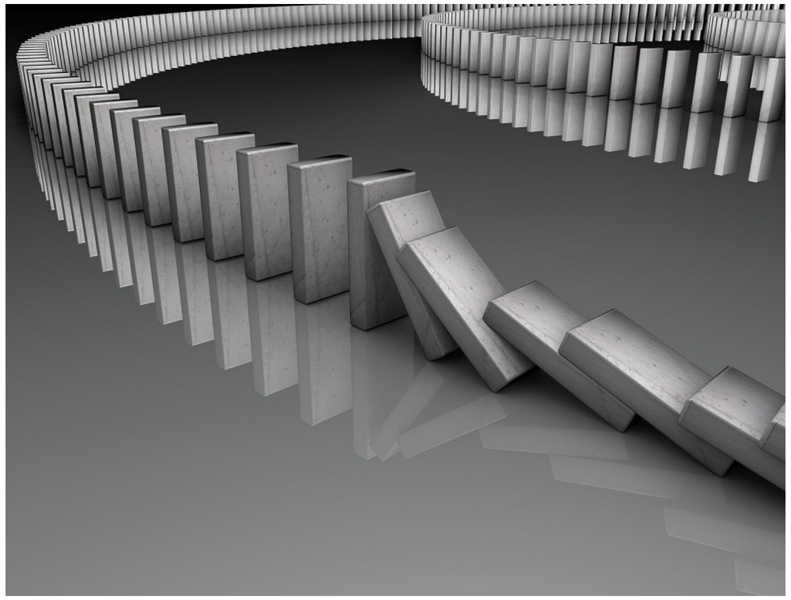
However;
Proving that a recurrence relation is satisfied by a closed expression
is something we will now learn how to do…
through *induction*…

Mathematical (full) induction
is a proof technique

that can be used to prove that some property holds for (set of of) natural numbers

`later..we will generalize all this to induction over structures!`

Mathematical induction



ℕ - natural numbers

Trying to check if some property P holds for all numbers.

(i) *base case*
ensure P(0) holds (is true)

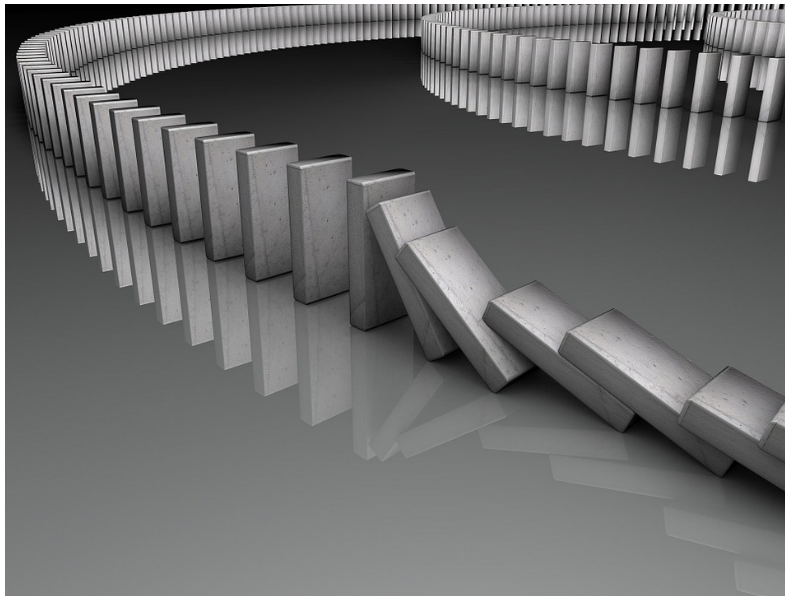
(ii) *inductive step (step case)*
$\underbrace{\textit{if } P(k) \textit{ holds,}}_{\textbf{\textit{induction hypothesis} or \textit{inductive hypothesis}}}$ *then so does P(k+1) (for all k)*

$$[P(0) \wedge \forall n\ (P(n) \Rightarrow P(n+1))] \Rightarrow \forall n\ P(n)$$

Figure credit: https://pixabay.com/illustrations/domino-game-falling-communication-163523/

Mathematical induction

Intuition:
"chain reactions"



you want to make sure all the dominos fall…

so you check that all two neighbouring are close enough….

this ensures that … (ii) ***if*** *the* k^{th} *falls,* ***then*** *so does the* $(k+1)^{th}$

So… if (ii) is true… and (i) the first one falls…
(i)+(ii) mean all will fall

Figure credit: https://pixabay.com/illustrations/domino-game-falling-communication-163523/

Example

Claim: $5^n - 2^n$ **is divisible by 3, for all** n**.**

(i) Base case

(ii) Inductive step

Example

Claim: $5^n - 2^n$ is divisible by 3, for all n.

(i) Base case $5 - 2 = 3$ ✓

(ii) Inductive step

NEED TO SHOW: IF $3 \mid 5^n - 2^n$ then $3 \mid 5^{n+1} - 2^{n+1}$

$$5^{n+1} - 2^{n+1} = 5 \times 5^n - 2^n \times 2 =$$

$$= 3 \times 5^n + 2 \times 5^n - 2 \times 2^n = \underbrace{3 \times 5^n}_{\text{div by 3}} + 2 \times (\underbrace{5^n - 2^n}_{\text{by assumption}})$$

Another example

Claim: $\sum_{i=1}^{n} i = n(n+1)/2$

(i) Base case

(ii) Inductive step

Another example

Claim: $\sum_{i=1}^{n} i = \mathbf{n(n+1)/2}$

(i) Base case $n=1 \Rightarrow 1 = \frac{1\times(1+1)}{2} = 1$ ✓

(ii) Inductive step

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

$$\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1) = \frac{n(n+1)}{2} + (n+1) = \frac{n(n+1)+2(n+1)}{2} = \frac{(n+2)(n+1)}{2} \quad \square$$

Use induction to solve expressions:

e.g. sum of squares $\sum_{k=1}^{n} k^2$

Sniff out solution

$$
\begin{aligned}
1 &= 1 = 1\cdot2\cdot3 / 6 \\
1+4 &= 5 = 2\cdot3\cdot5 / 6 \\
1+4+9 &= 14 = 3\cdot4\cdot7 / 6 \\
1+4+9+16 &= 30 = 4\cdot5\cdot9 / 6 \\
1+4+9+16+25 &= 55 = 5\cdot6\cdot11 / 6
\end{aligned}
$$

Guess: $$6 \cdot \sum_{k=1}^{n} k^2 = n(n+1)(2n+1)$$

Prove:

$$6 \cdot \sum_{k}^{n} k^2 = n(n+1)(2n+1)$$

(i) Base case

$6 \cdot (1^2 + 2^2 \cdots n^2 + (n+1)^2)$

(ii) Inductive step

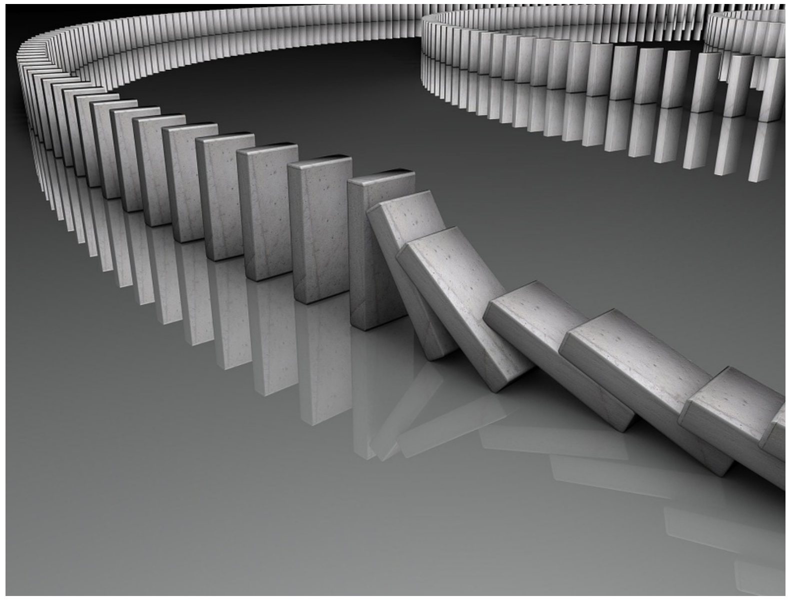
$= 6 \cdot \sum_{k=1}^{n} k^2 + 6(n+1)^2$

$$6 \cdot \sum_{k}^{n+1} k^2 = 6 \cdot \sum_{k}^{n} k^2 + 6 \cdot (n+1)^2 = n(n+1)(2n+1) + 6(n+1)^2 =$$

$$= (n+1)[n(2n+1) + 6(n+1)] = (n+1)[2n^2 + n + 6n + 6] =$$

$$= (n+1)(2n^2 + 7n + 6) = (n+1)(n+2)(2n+3)$$

Mathematical induction - subset



$\mathbb{N}$ - *natural numbers*

Trying to check if some property P holds for subset of all numbers $n \geq n_0$.

(i) *base case*
ensure $P(n_0)$ holds

(ii) *inductive step (step case)*
if $P(k)$ holds, then so does $P(k+1)$ (for all $k \geq n_0$)

Figure credit: https://pixabay.com/illustrations/domino-game-falling-communication-163523/

Mathematical induction - equivalent formulation (JUST FYI)

ℕ - natural numbers

Trying to check if some property P holds for all numbers.

(i) *<u>base case</u>*
ensure P(0) holds (is true)

(ii) *<u>inductive step (step case)</u>*
prove that if P(k) holds for all value k<n then it holds for n.

(A) => (B)

Recurrence relations: proving a debt

- $t_0 = 5,\ t_1 = 6$
- $t_n = t_{n-1} + 6t_{n-2} - 12n + 8, \quad n \geq 2$

claim of closed form expression: $t_n = 3^n + (-2)^n + 2n + 3$

(i) Base case(s)

$$(n = 0) \quad 1 + 1 + 3 = 5; (n = 1) \quad 3 - 2 + 2 + 3 = 6$$

(ii) Inductive step

$$t_{n+1} = t_n + t_{n-1} + 12(n + 1) + 8$$

Recurrence relations

- $t_0 = 5, \ t_1 = 6$
- $t_n = t_{n-1} + 6t_{n-2} - 12n + 8, \quad n \geq 2$

$$t_n = 3^n + (-2)^n + 2n + 3$$

Recurrence relations

- $t_0 = 5,\ t_1 = 6$
- $t_n = t_{n-1} + 6t_{n-2} - 12n + 8, \quad n \geq 2$

$$t_n = 3^n + (-2)^n + 2n + 3$$

$$t_{n+1} = t_n + 6t_{n-1} - 12(n+1) + 8$$

$$= 3^n + (-2)^n + 2n + 3 + 6\left(3^{n-1} + (-2)^{n-1} + 2(n-1) + 3\right) - 12(n+1) + 8$$

$$= \left(2\times 3^n + 3^n\right) + (-2)^n - 3(-2)^n + 2n + 6\times 2(n-1) - 12(n+1) + 3 + 3\times 6 + 8$$

$$= 3^{n+1} + (-2)(-2)^n + 2n + 12n - 12 - 12n - 12 + 3 + 3\times 6 + 8 = 3^{n+1} + (-2)^{n+1} + 2n + 3.$$

$-24 \quad \leftarrow 27$

Practice:

Prove: $1 + \sum_{k=1}^{n} (k \times k!) = (n+1)!, \quad \textbf{for } n \geq 1.$

Practice:

Prove: $1 + \sum_{k=1}^{n} (k \times k!) = (n+1)!$, **for** $n \geq 1$.

On board

Practice:

$$\text{Prove: } 1 + \sum_{k=1}^{n} (k \times k!) = (n+1)!, \quad \textbf{for } n \geq 1.$$

Practice:

Prove: $1 + \sum_{k=1}^{n} (k \times k!) = (n+1)!$, **for** $n \geq 1$.