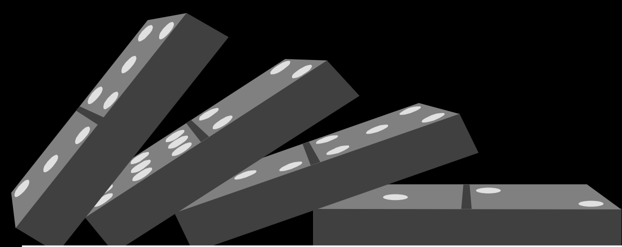
15-251

Great Theoretical Ideas in Computer Science

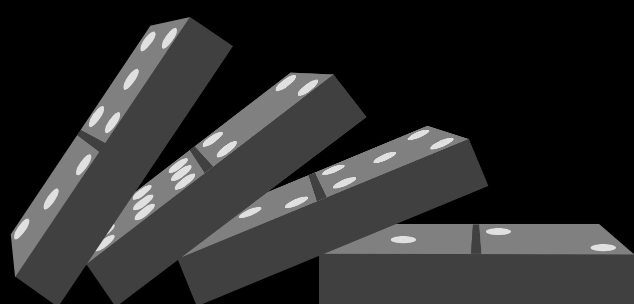


Inductive Reasoning

Lecture 3 (September 4, 2007)

Dominoes

Domino Principle: Line up any number of dominos in a row; knock the first one over and they will all fall



Dominoes Numbered 1 to n

F_k = "The k[th] domino falls"

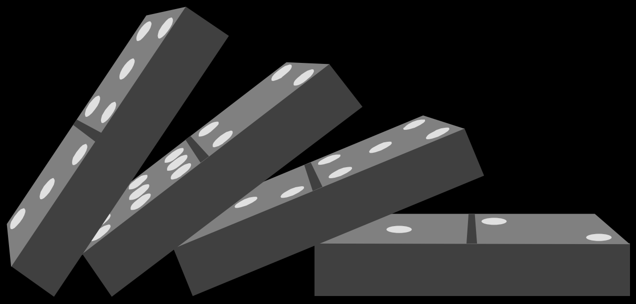
If we set them up in a row then each one is set up to knock over the next:

For all $1 \leq k < n$:

$$F_k \Rightarrow F_{k+1}$$

$F_1 \Rightarrow F_2 \Rightarrow F_3 \Rightarrow \ldots$

$F_1 \Rightarrow$ All Dominoes Fall



Standard Notation

"for all" is written "$\forall$"

Example:

For all k>0, P(k) = $\forall$ k>0, P(k)

Dominoes Numbered 1 to n

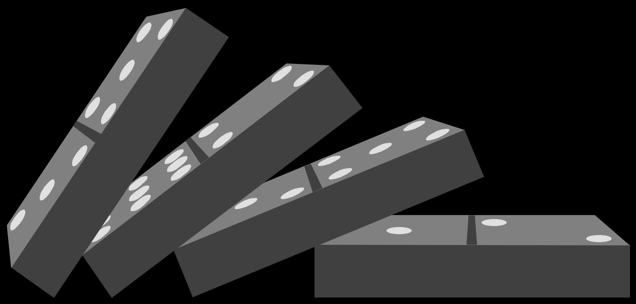
F_k = "The k^{th} domino falls"

$\forall k, 0 \leq k < n-1:$

$$F_k \Rightarrow F_{k+1}$$

$F_0 \Rightarrow F_1 \Rightarrow F_2 \Rightarrow \ldots$

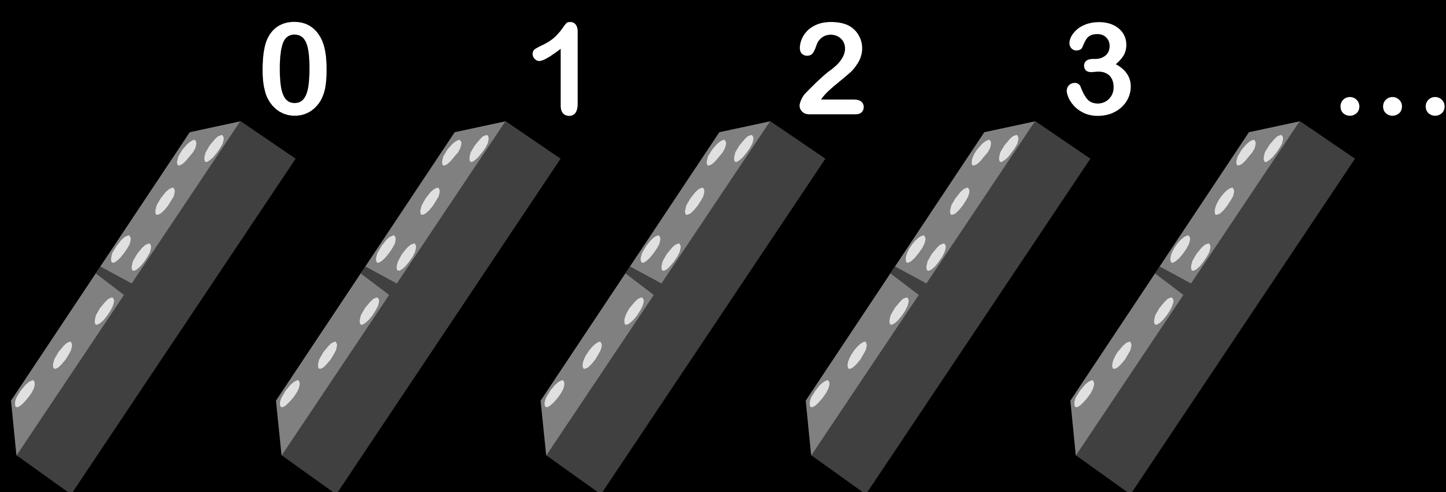
$F_0 \Rightarrow$ All Dominoes Fall

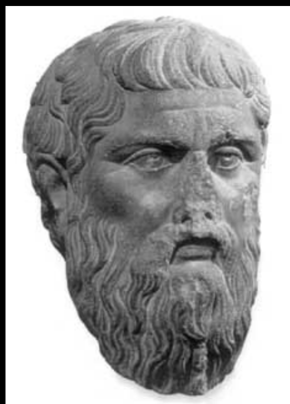


The Natural Numbers

$\mathbb{N} = \{ 0, 1, 2, 3, \ldots \}$

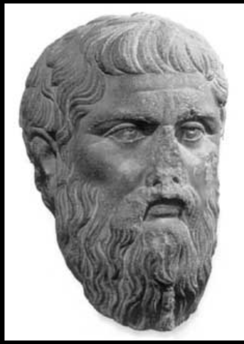
One domino for each natural number:





Plato: The Domino Principle works for an infinite row of dominoes

Aristotle: Never seen an infinite number of anything, much less dominoes.

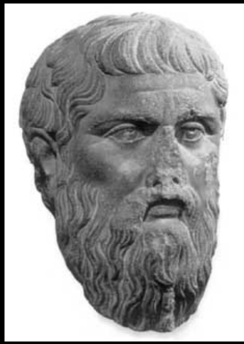


Plato's Dominoes

One for each natural number

Theorem: An infinite row of dominoes, one domino for each natural number.

Knock over the first domino and they all will fall



Plato’s Dominoes

One for each natural number

Theorem: An infinite row of dominoes, one domino for each natural number.

Knock over the first domino and they all will fall

Proof:

Suppose they don’t all fall. Let $k > 0$ be the lowest numbered domino that remains standing.
Domino $k-1 \geq 0$ did fall, but $k-1$ will knock over domino k. Thus, domino k must fall and remain standing.
Contradiction.

Mathematical Induction

statements proved instead of dominoes fallen

Infinite sequence of dominoes

F_k = "domino k fell"

Infinite sequence of statements: S_0, S_1, …

F_k = "S_k proved"

Establish: 1. F_0

2. For all k, $F_k \Rightarrow F_{k+1}$

Conclude that F_k is true for all k

Inductive Proofs

To Prove $\forall k \in \mathbb{N}, S_k$

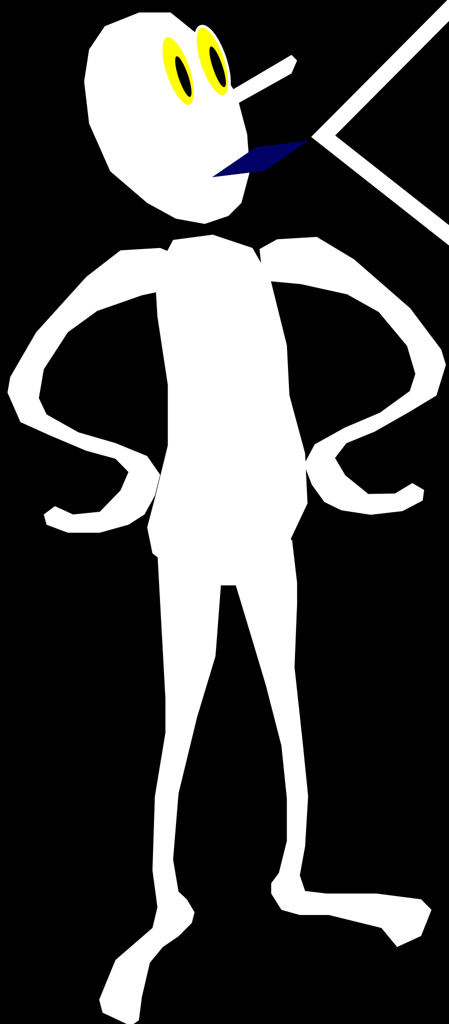
Establish "Base Case": S_0

Establish that $\forall k, S_k \Rightarrow S_{k+1}$

Induction hypoth

$\forall k, S_k \Rightarrow S_{k+1}$

- Assume hypothetically that S_k for any particular k;
- Conclude that S_{k+1}



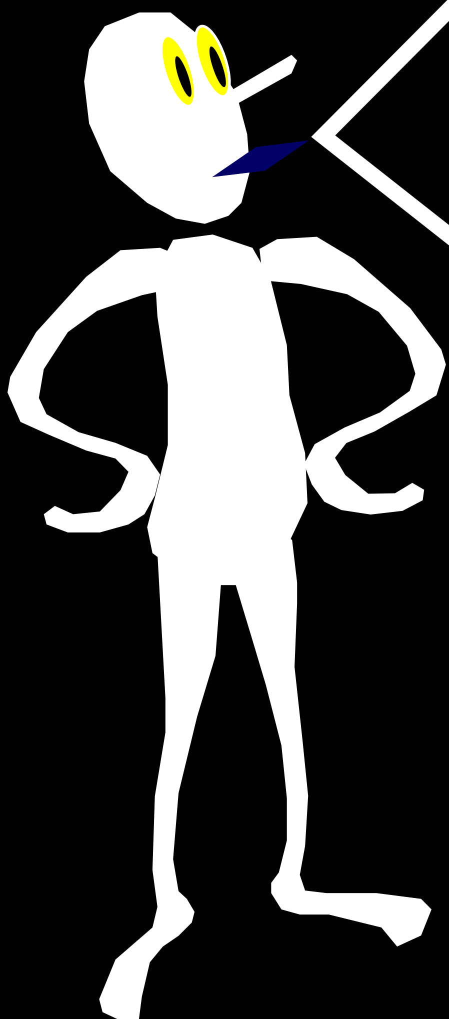
Theorem?

The sum of the first n **odd** numbers is n^2

$$1 = 1$$

$$1+3 = 2^2$$

$$1+3+5 = 9 = 3^2$$

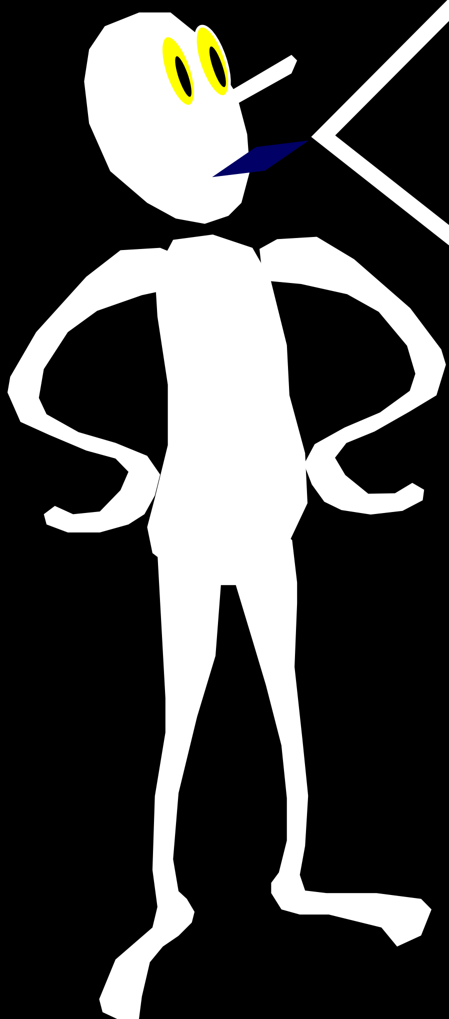


Theorem?

The sum of the first n odd numbers is n^2

Check on small values:

1	= 1
1+3	= 4
1+3+5	= 9
1+3+5+7	= 16



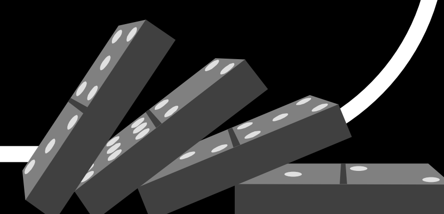
Theorem?

The sum of the first n **odd** numbers is n^2

The k^{th} odd number is $(2k – 1)$, when $k > 0$

S_n is the statement that:

“$1+3+5+(2k-1)+\ldots+(2n-1) = n^2$”



Establishing that $\forall n \geq 1\ S_n$

S_n = "$1 + 3 + 5 + (2k-1) + \ldots + (2n-1) = n^2$"

Base Case: S_1 = " $1 = 1^2$ " ✓

$\forall k\ S_k \Rightarrow S_{k+1}$

I.H. S_k = " $1 + 3 + \ldots + (2k-1) = k^2$ "

Induction Step.

S_{k+1} $\underbrace{1 + 3 + 5 + \ldots + (2k-1)} + (2k+1)$

$= k^2 + (2k+1)$ by I.H.

$= (k+1)^2$

Establishing that $\forall n \geq 1\ S_n$

S_n = "$1 + 3 + 5 + (2k-1) + . . +(2n-1) = n^2$"

Establishing that $\forall n \geq 1\ S_n$

S_n = “$1 + 3 + 5 + (2k-1) + . . +(2n-1) = n^2$”

Base Case: S_1

Domino Property:

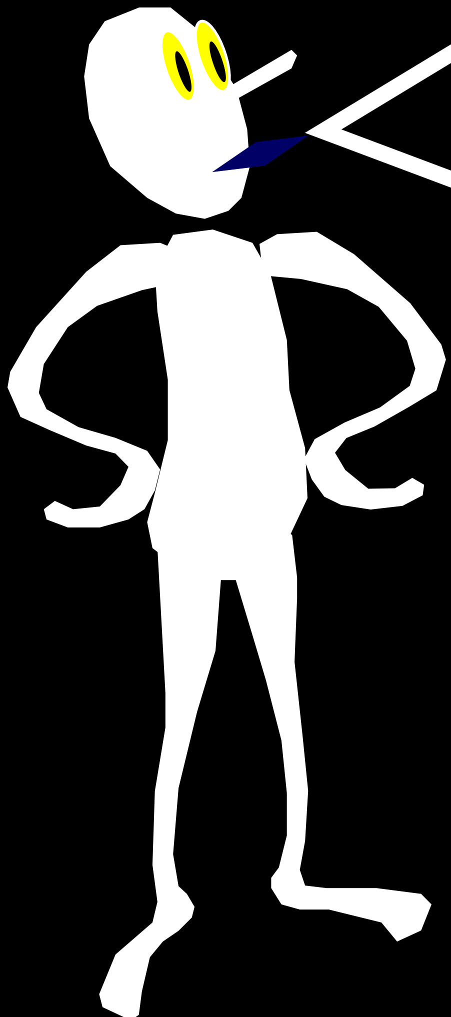
Assume “Induction Hypothesis”: S_k

That means:

$1+3+5+\ldots+ (2k-1) \qquad = k^2$

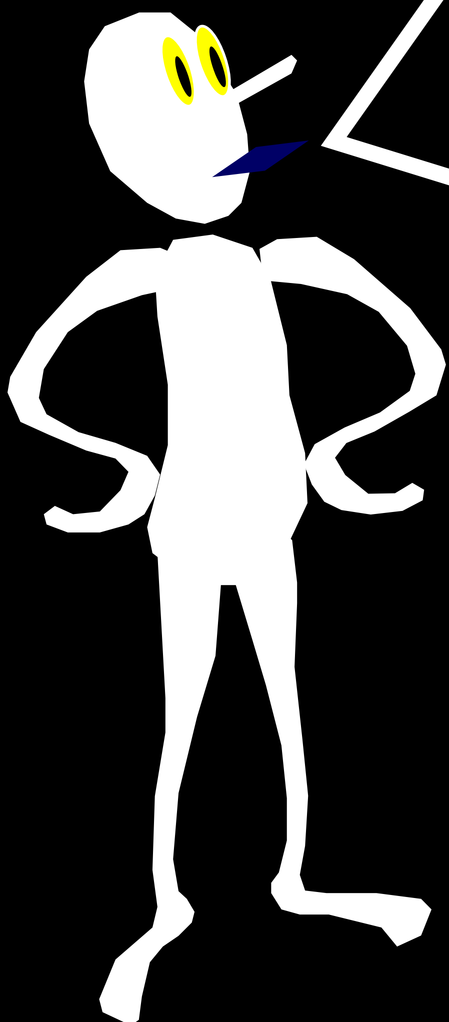
$1+3+5+\ldots+ (2k-1)+(2k+1) \qquad = k^2 +(2k+1)$

Sum of first k+1 odd numbers = $(k+1)^2$



Theorem

The sum of the first n odd numbers is n^2



Primes:

A natural number n > 1 is a **prime** if it has no divisors besides 1 and itself

Note: 1 is not considered prime

Theorem?

Every natural number > 1 can be factored into primes

S_n = "n can be factored into primes"

Base case:

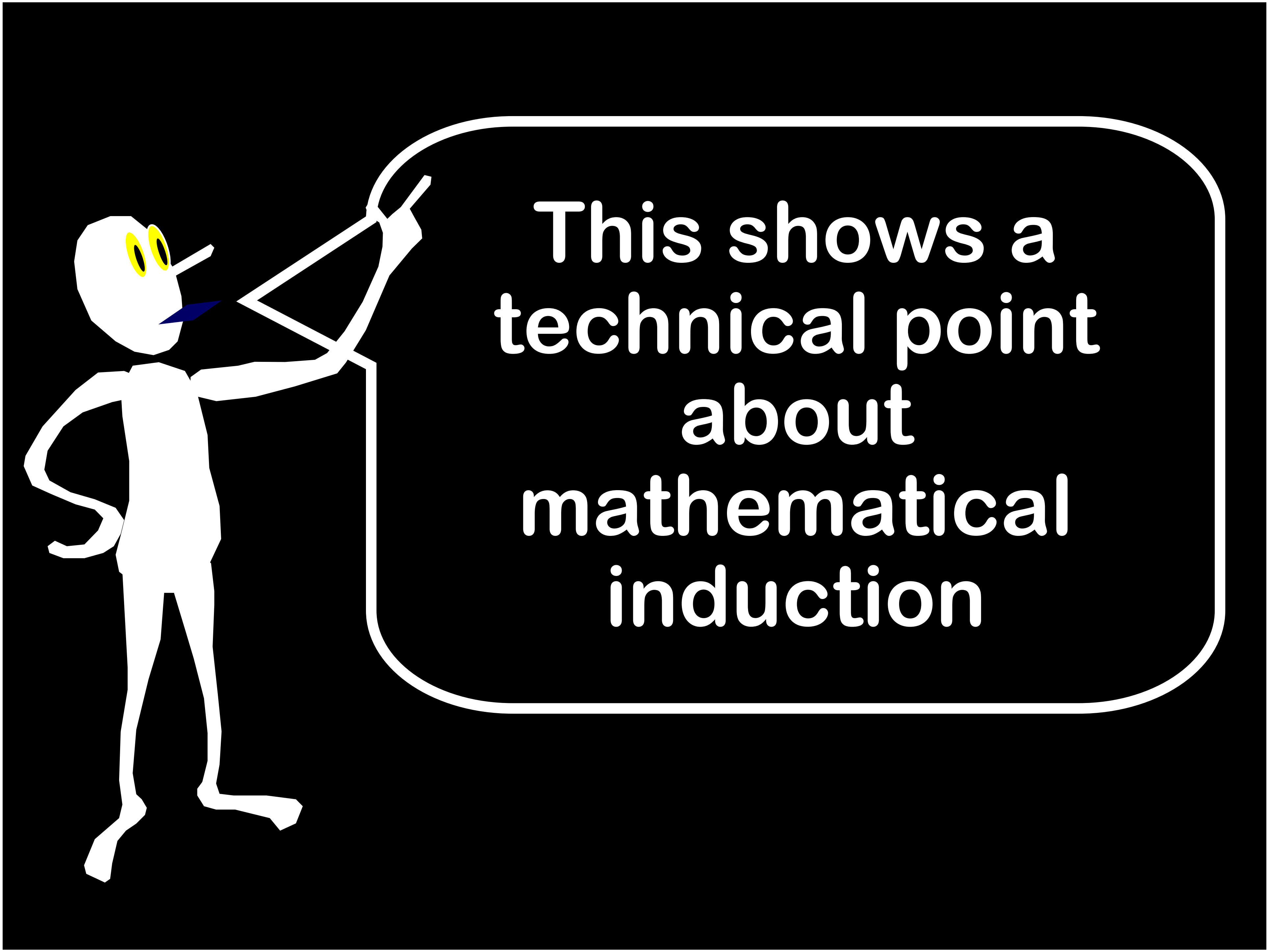
2 is prime $\Rightarrow$ S_2 is true

How do we use the fact:

S_{k-1} = "k-1 can be factored into primes"

to prove that:

S_k = "k can be factored into primes"


This shows a technical point about mathematical induction

A different approach:

Assume 2,3,…,k-1 **all** can be factored into primes

Then show that k can be factored into primes

Theorem?

Every natural number > 1 can be factored into primes

S_n = "n can be written as product of primes"

Base Case: S_2 ✓

I.H.: Suppose $S_2, S_3 \ldots S_{k-1}$ all true

S_k: either k is a prime ✓

or $k = a.b$ with $a, b < k$

$S_a \Rightarrow$ a can be written as product of primes

$S_b \Rightarrow$ b ——— — — —

k = can be written as product of primes.

All Previous Induction

To Prove $\forall k, S_k$

Establish Base Case: S_0

Establish Domino Effect:

Assume $\forall j<k, S_j$

use that to derive S_k

ion

Also called “Strong Induction”

Establish Domino Effect:

Assume $\forall j<k$, S_j

use that to derive S_k

"All Previous" Induction
Repackaged As Standard Induction

Establish Base Case: S_0

Establish Domino Effect:

Let k be any number
Assume $\forall j<k, S_j$

Prove S_k

Define $T_i = \forall j \leq i, S_j$

Establish Base Case T_0

Establish that $\forall k, T_k \Rightarrow T_{k+1}$

Let k be any number
Assume T_{k-1}

Prove T_k


And there are more ways to do inductive proofs

Method of Infinite Descent

Pierre de Fermat

$x^3 + y^3 = z^3$

$x^4 + y^4 = z^4$ $x^n + y^n = z^n$

Show that for any counter-example you find a smaller one

If a counter-example exists there would be an infinite sequence of smaller and smaller counter-examples

Theorem:

Every natural number > 1 can be factored into primes

Let **n** be a counter-example

Hence **n** is not prime, so **n = ab**

If both **a** and **b** had prime factorizations, then **n** would too

Thus **a** or **b** is a smaller counter-example

Theorem:

Every natural number > 1 can be factored into primes

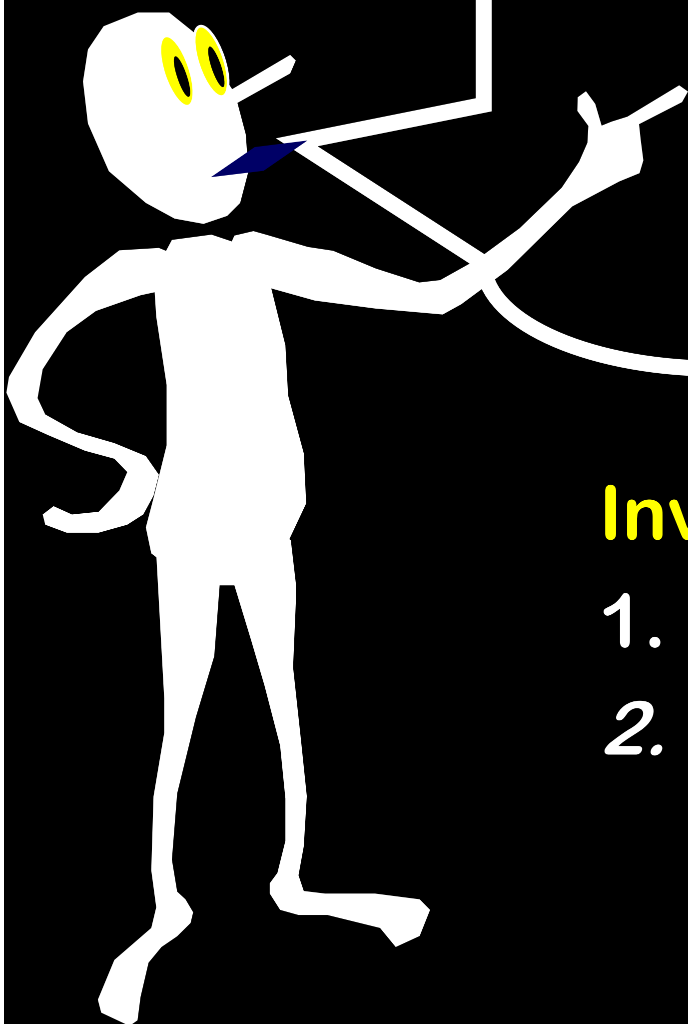
Let n be ~~a~~ the smallest / least counter-example

Hence n is not prime, so n = ab

If both a and b had prime factorizations, then n would too

Thus a or b is a smaller counter-example

$\Rightarrow$ contradiction



Yet another way of packaging inductive reasoning is to define "invariants"

Invariant (n):

1. Not varying; constant.

2. Mathematics. Unaffected by a designated operation, as a transformation of coordinates.

Invariant (n):

3. *Programming.* A rule, such as the ordering of an ordered list, that applies throughout the life of a data structure or procedure. Each change to the data structure maintains the correctness of the invariant

Invariant Induction

Suppose we have a time varying world state: W_0, W_1, W_2, …

Each state change is assumed to come from a list of permissible operations. We seek to prove that statement S is true of all future worlds

Argue that S is true of the initial world

Show that if S is true of some world – then S remains true after one permissible operation is performed

Odd/Even Handshaking Theorem

At any party at any point in time define a person's parity as ODD/EVEN according to the number of hands they have shaken

Statement:
The number of people of odd parity must be even

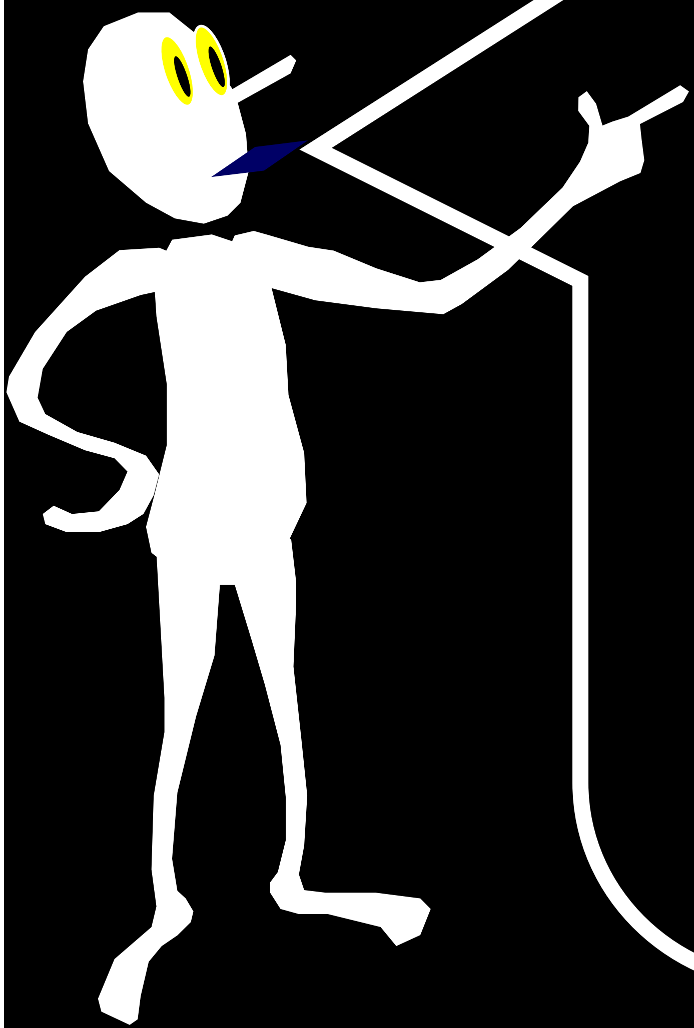
Statement: The number of people of odd parity must be even

Initial case: Zero hands have been shaken at the start of a party, so zero people have odd parity

Invariant Argument:

If 2 people of the same parity shake, they both change and hence the odd parity count changes by 2 – and remains even

If 2 people of different parities shake, then they both swap parities and the odd parity count is unchanged



Inductive reasoning is the high level idea

"Standard" Induction

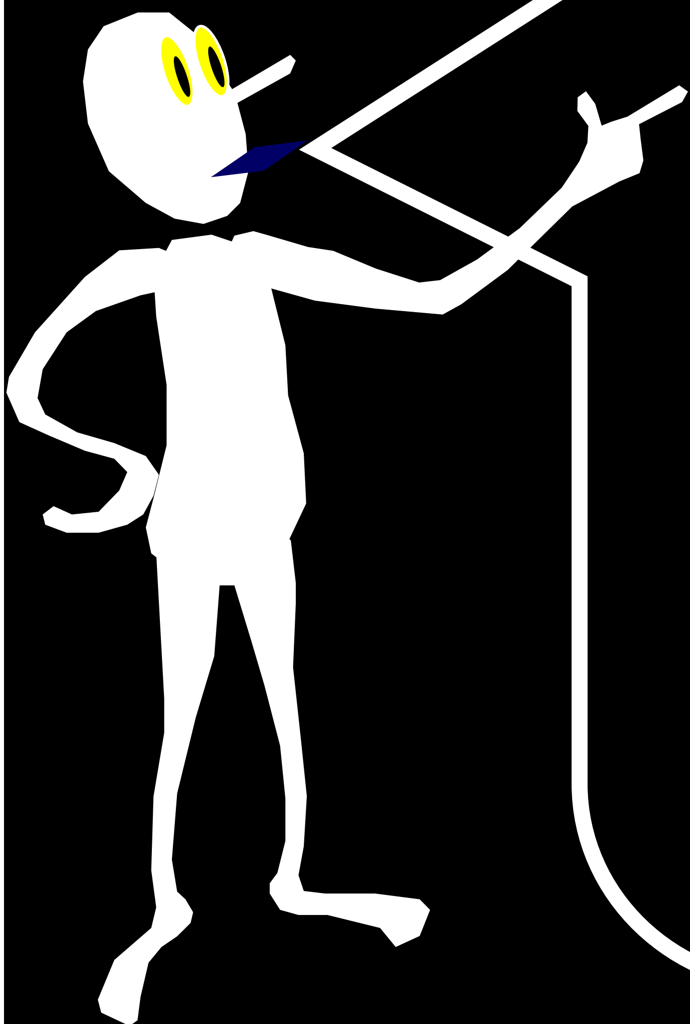
"All Previous" Induction

"Least Counter-example"

"Invariants"

all just

different packaging



Induction is also how we can define and construct our world

So many things, from buildings to computers, are built up stage by stage, module by module, each depending on the previous stages

Inductive Definition

Example

Initial Condition, or Base Case:

$F(0) = 1$

Inductive definition of the powers of 2!

Inductive Rule:

For $n > 0$, $F(n) = F(n-1) + F(n-1)$

n	0	1	2	3	4	5	6	7
F(n)	1	2	4	8	16	32	64	128

Leonardo Fibonacci

In 1202, Fibonacci proposed a problem about the growth of rabbit populations

Rabbit Reproduction

A rabbit lives forever

The population starts as single newborn pair

Every month, each productive pair begets a new pair which will become productive when they are 2 months old

F_n= # of rabbit pairs at the beginning of the n^{th} month

month	1	2	3	4	5	6	7
rabbits	1	1	2	3	5	8	13

Fibonacci Numbers

month	1	2	3	4	5	6	7
rabbits	1	1	2	3	5	8	13

Stage 0, Initial Condition, or Base Case:

Fib(1) = 1; Fib (2) = 1

Inductive Rule:

For $n>3$, Fib(n) = Fib(n-1) + Fib(n-2)

Recurrences

Example

$T(1) = 1$

$T(n) = 4T(n/2) + n$

Notice that T(n) is inductively defined only for positive powers of 2, and undefined on other values

$T(1) = 1$ $T(2) = 6$ $T(4) = 28$ $T(8) = 120$

Guess a closed-form formula for T(n)

Guess: $G(n) = 2n^2 - n$

Inductive Proof of Equivalence

$$G(n) = 2n^2 - n$$

$$T(1) = 1$$

$$T(n) = 4T(n/2) + n$$

Inductive Proof of Equivalence

Base Case: $G(1) = 1$ and $T(1) = 1$

Induction Hypothesis:

$T(x) = G(x)$ for $x < n$

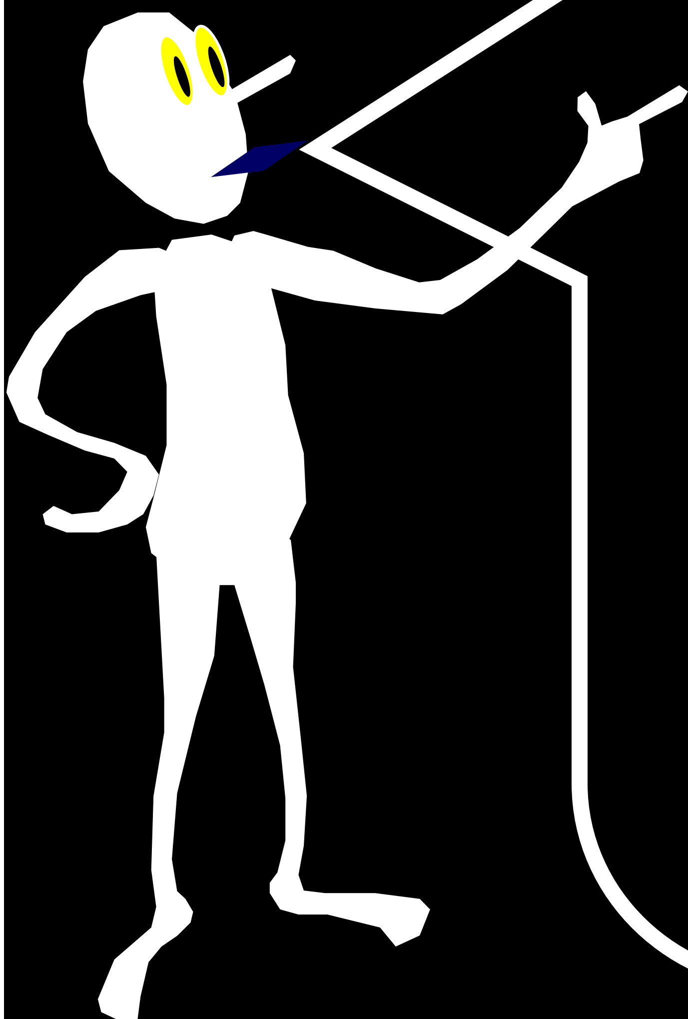
Hence: $T(n/2) = G(n/2) = 2(n/2)^2 – n/2$

$$
\begin{aligned}
T(n) &= 4\,T(n/2) + n \\
&= 4\,G(n/2) + n \\
&= 4\,[2(n/2)^2 – n/2] + n \\
&= 2n^2 – 2n + n \\
&= 2n^2 – n \\
&= G(n)
\end{aligned}
$$

$G(n) = 2n^2 - n$

$T(1) = 1$

$T(n) = 4T(n/2) + n$



We inductively proved the assertion that $G(n) = T(n)$

Giving a formula for T with no recurrences is called “solving the recurrence for T”

Technique 2

Guess Form, Calculate Coefficients

$T(1) = 1$, $T(n) = 4\,T(n/2) + n$

Guess: $T(n) = an^2 + bn + c$

for some a,b,c

Technique 2

Guess Form, Calculate Coefficients

$T(1) = 1, T(n) = 4\,T(n/2) + n$

Guess: $T(n) = an^2 + bn + c$

for some a,b,c

Calculate: $T(1) = 1$, so $a + b + c = 1$

$T(n) = 4\,T(n/2) + n$

$an^2 + bn + c = 4\,[a(n/2)^2 + b(n/2) + c] + n$

$= an^2 + 2bn + 4c + n$

$(b+1)n + 3c = 0$

Therefore: $b = -1 \quad c = 0 \quad a = 2$

$T(n) = 2n^2 - 1 \cdot n$

Inductive Definitions: some examples

The Lindenmayer Game

Alphabet: {a,b}

Start word: a

Productions Rules:

Sub(a) = ab   Sub(b) = a

NEXT(w_1 w_2 … w_n) =
Sub(w_1) Sub(w_2) … Sub(w_n)

Time 1: a

Time 2: ab

Time 3: aba

Time 4: abaab

Time 5: abaababa

How long are the strings at time n?

FIBONACCI(n)

Aristid Lindenmayer (1925-1989)

- 1968 Invents L-systems in Theoretical Botany

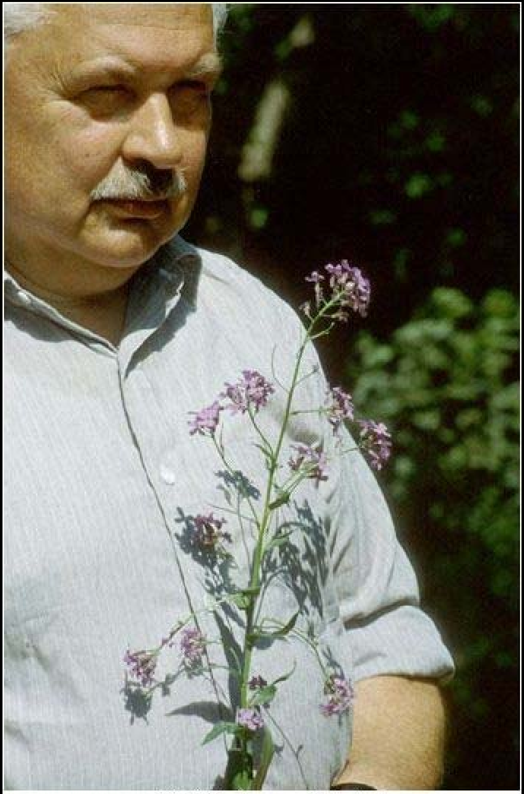
Time 1: a

Time 2: ab

Time 3: aba

Time 4: abaab

Time 5: abaababa



Aristid Lindenmayer 1925–1989

The Koch Game


Helge von Koch

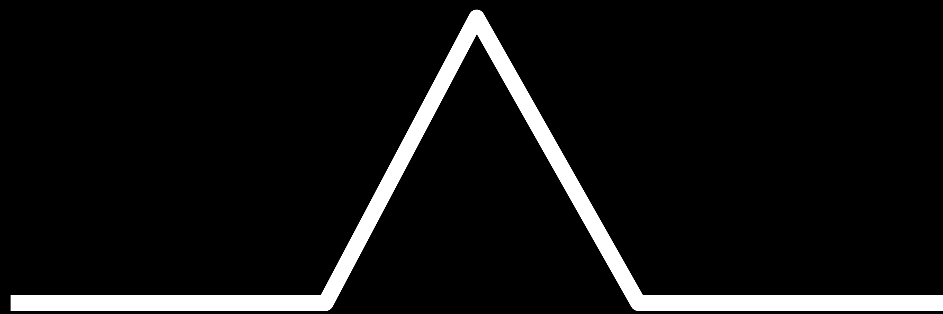
- **Alphabet:** $\Sigma = \{ F, +, - \}$
- **Start word:** F
- **Production Rules:**
- Sub(F) = F+F--F+F
- Sub(+) = +
- Sub(-) = -
- **NEXT**$(w_1\ w_2 \ldots w_n) = \text{Sub}(w_1)\ \text{Sub}(w_2) \ldots \text{Sub}(w_n)$

Gen 0: F

Gen 1: F+F--F+F

Gen 2: F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F

The Koch Game

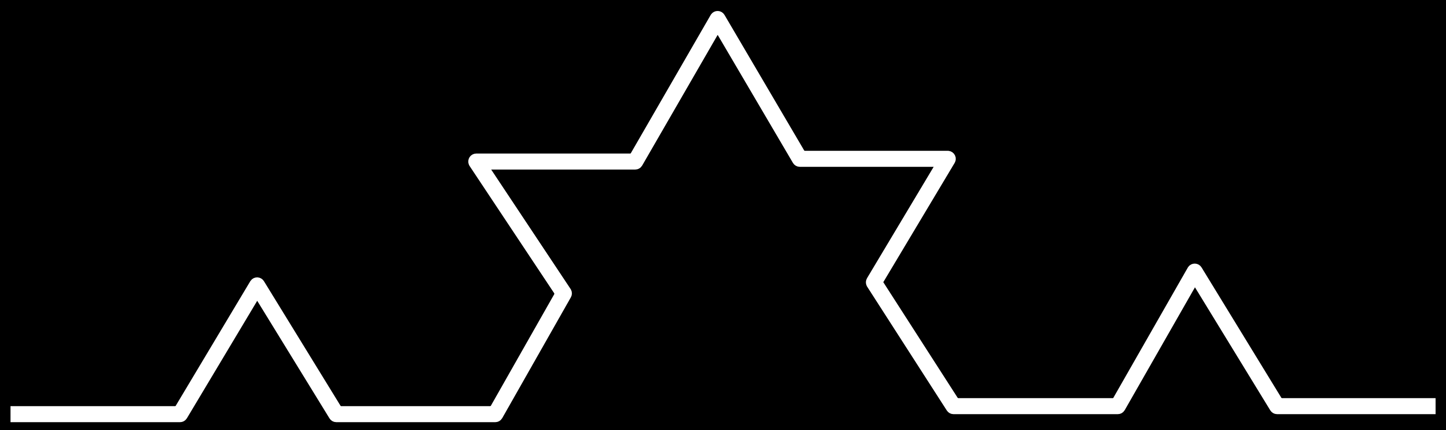


F+F--F+F

Visual representation:

F	draw forward one unit
+	turn 60 degree left
-	turn 60 degrees right

The Koch Game



F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F

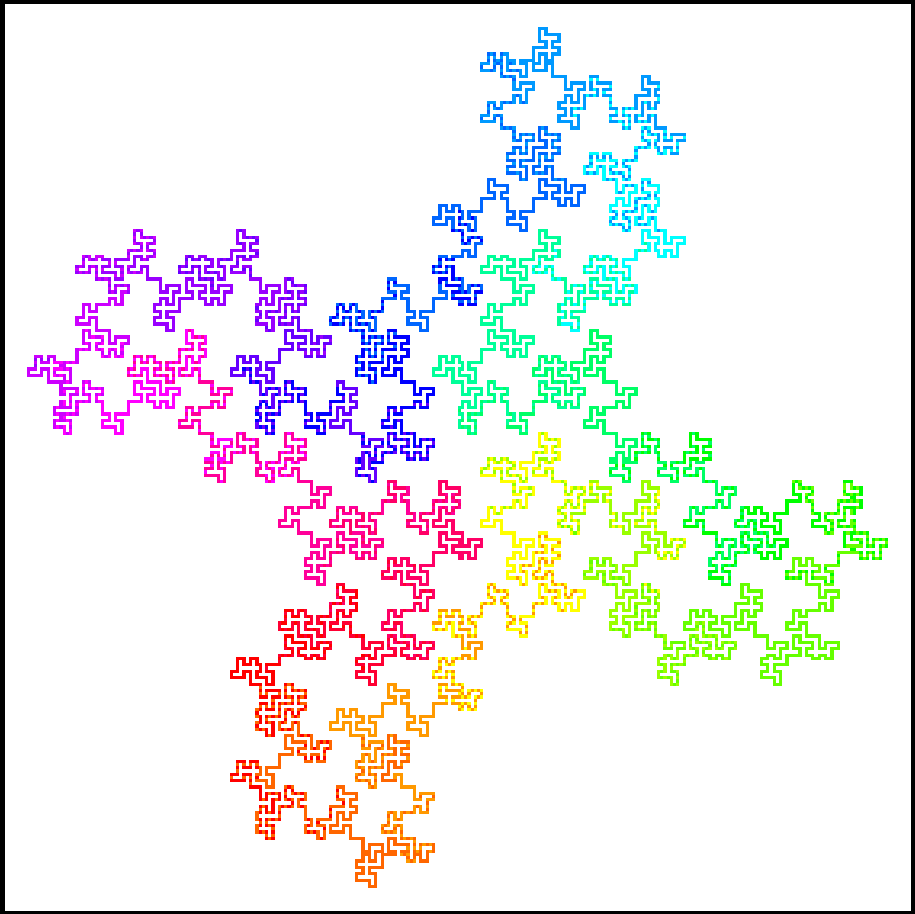
Visual representation:

F   draw forward one unit

+   turn 60 degree left

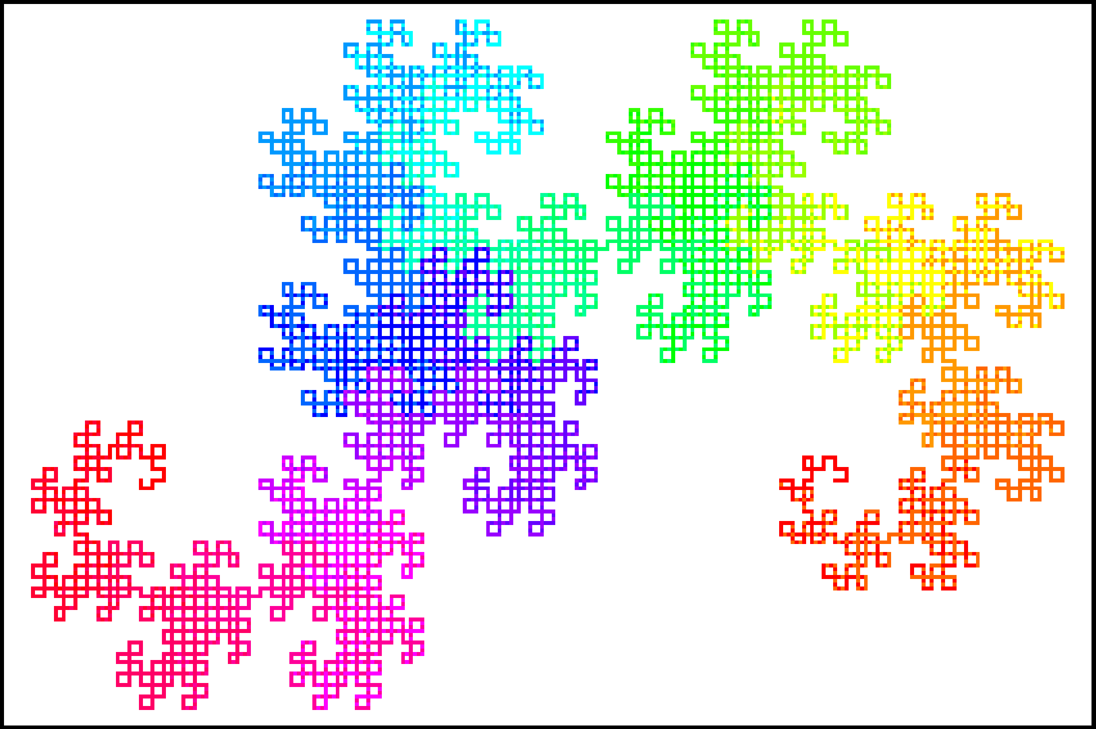
-      turn 60 degrees right

Koch Curve



Dragon Game

Sub(X) =X+YF+ Sub(Y) = -FX-Y



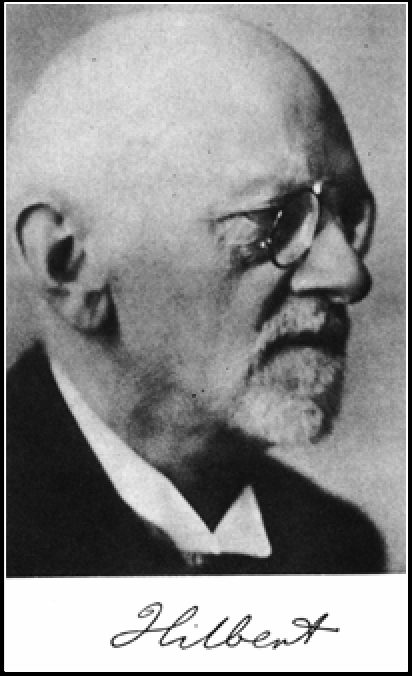
Dragon Curve

Hilbert Game

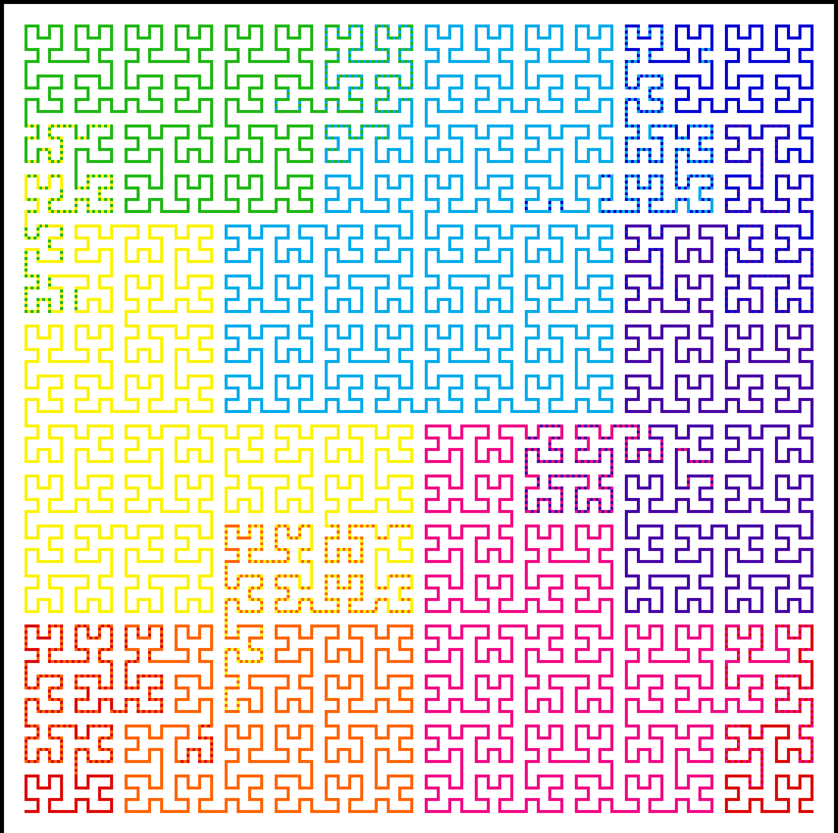
Sub(L)= +RF-LFL-FR+

Sub(R)= -LF+RFR+FL-

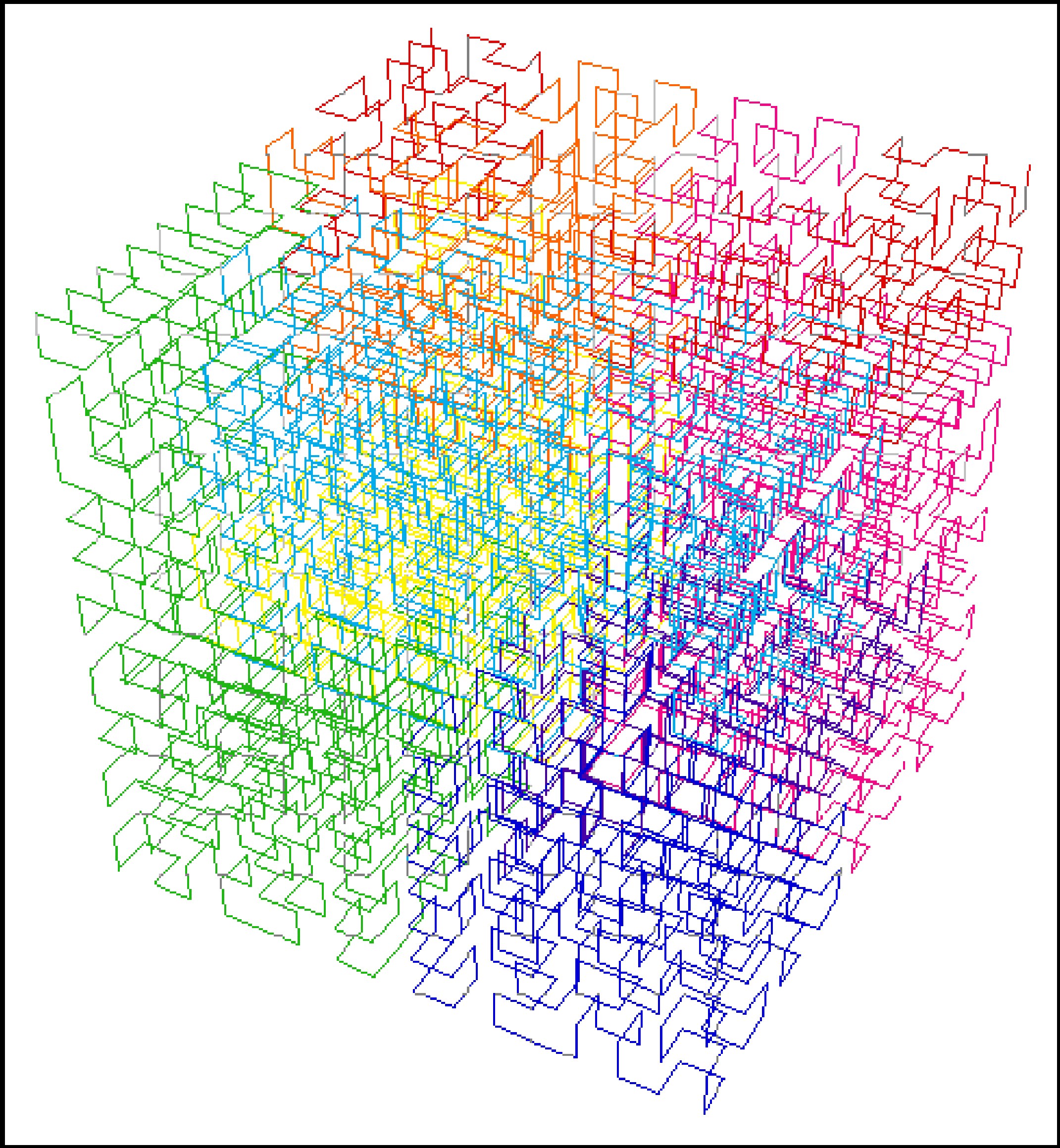
Make 90 degree turns instead of 60 degrees.



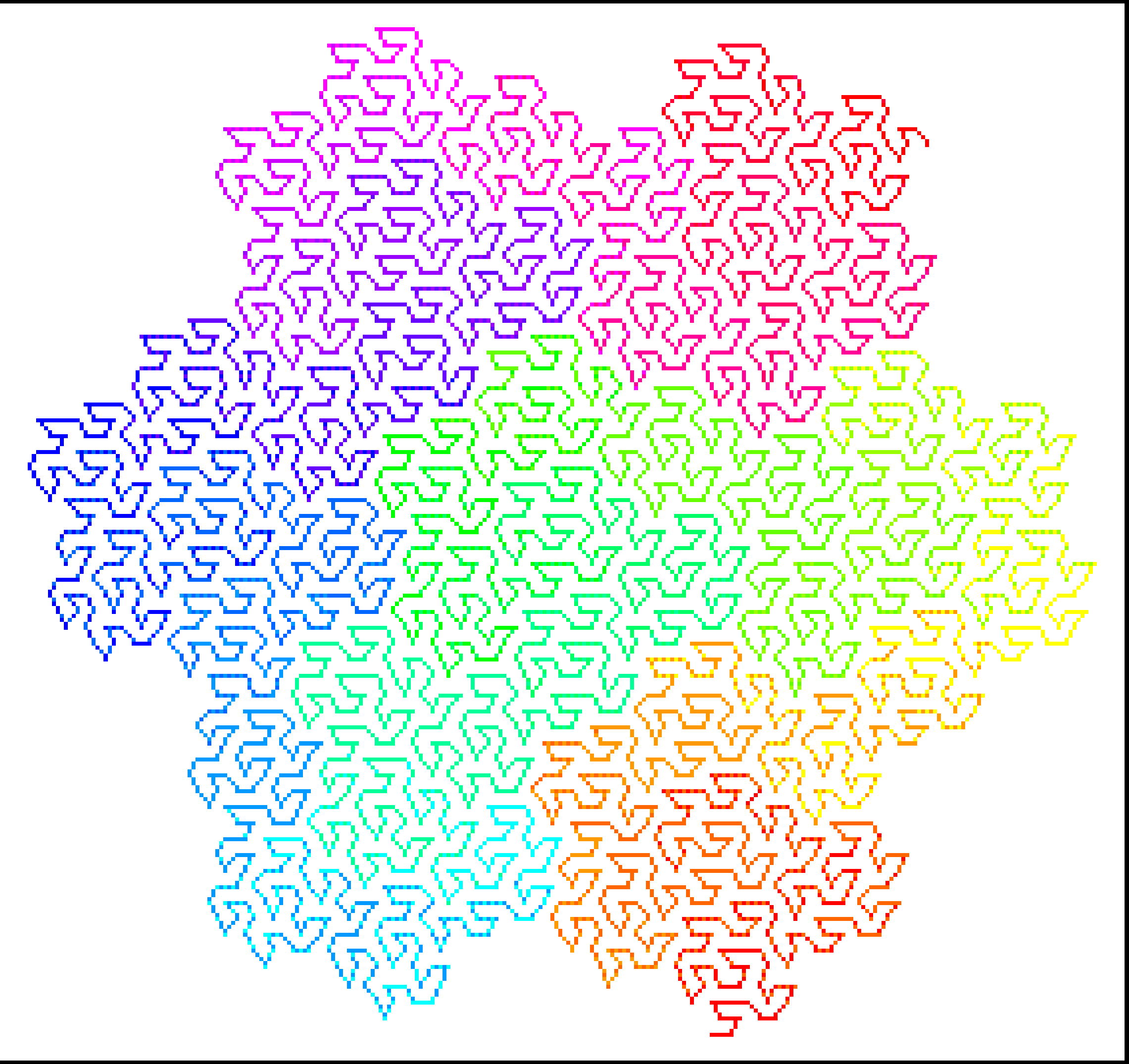
Hilbert Curve



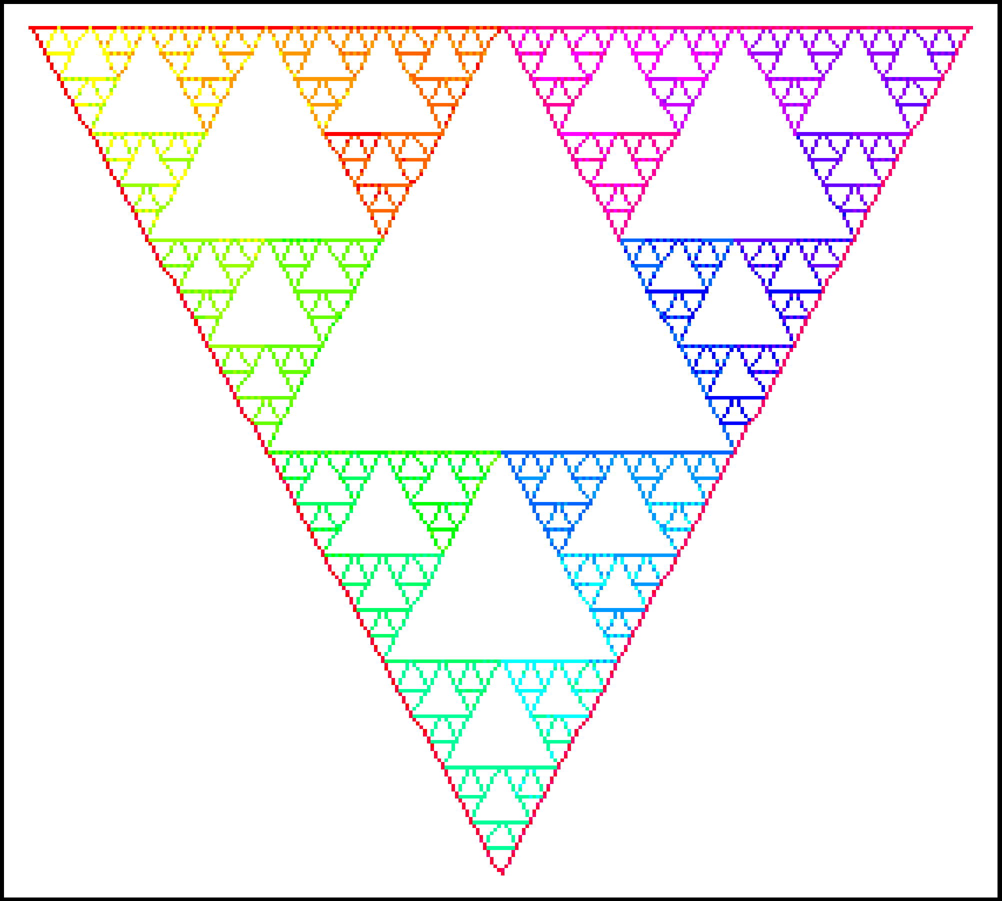
Hilbert's space filling curve



Peano's gossamer curve



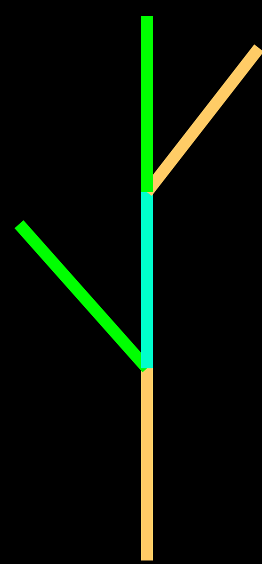
Sierpinski’s triangle



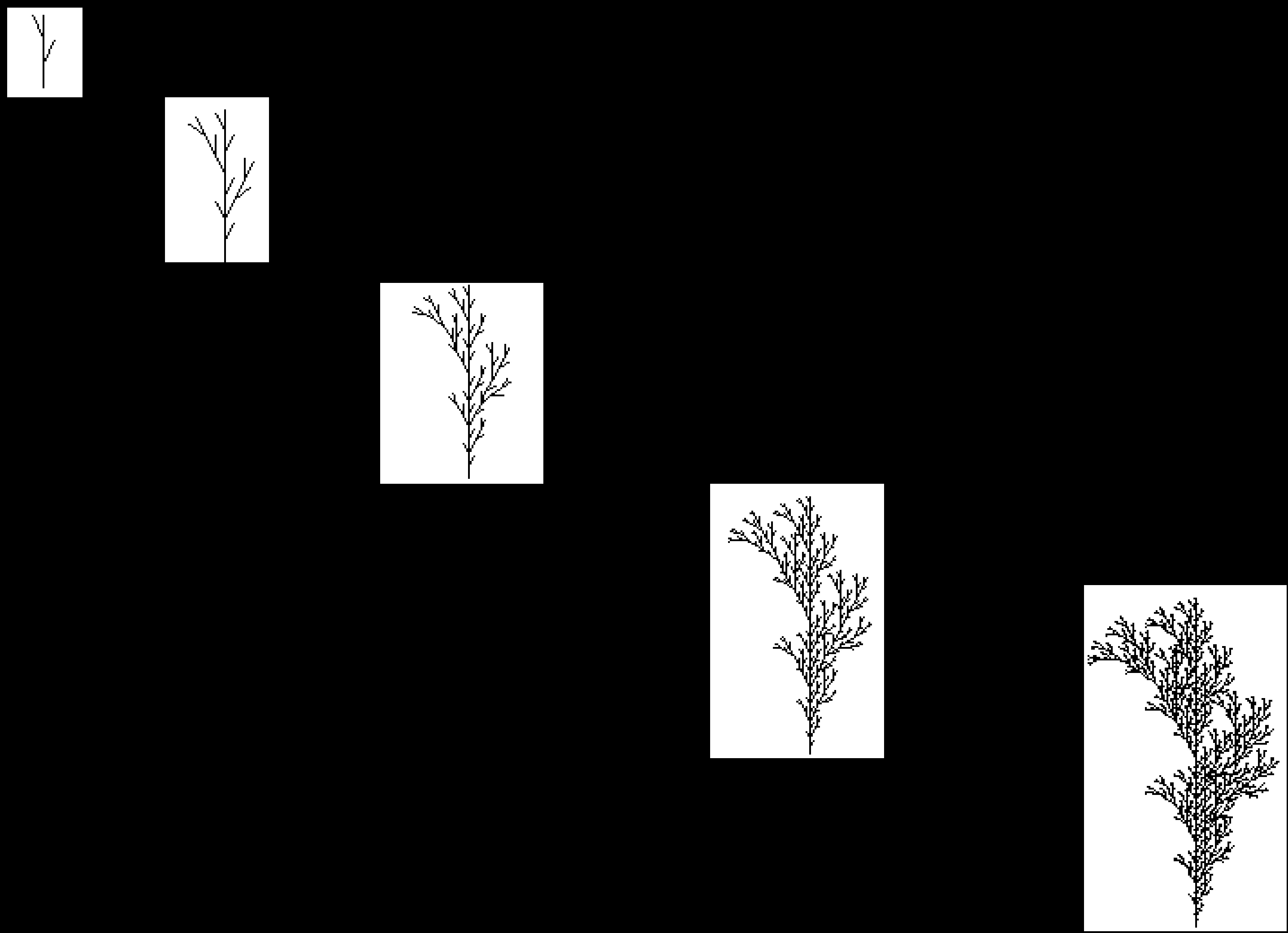
Lindenmayer 1968

Sub(F) = F[-F]F[+F][F]

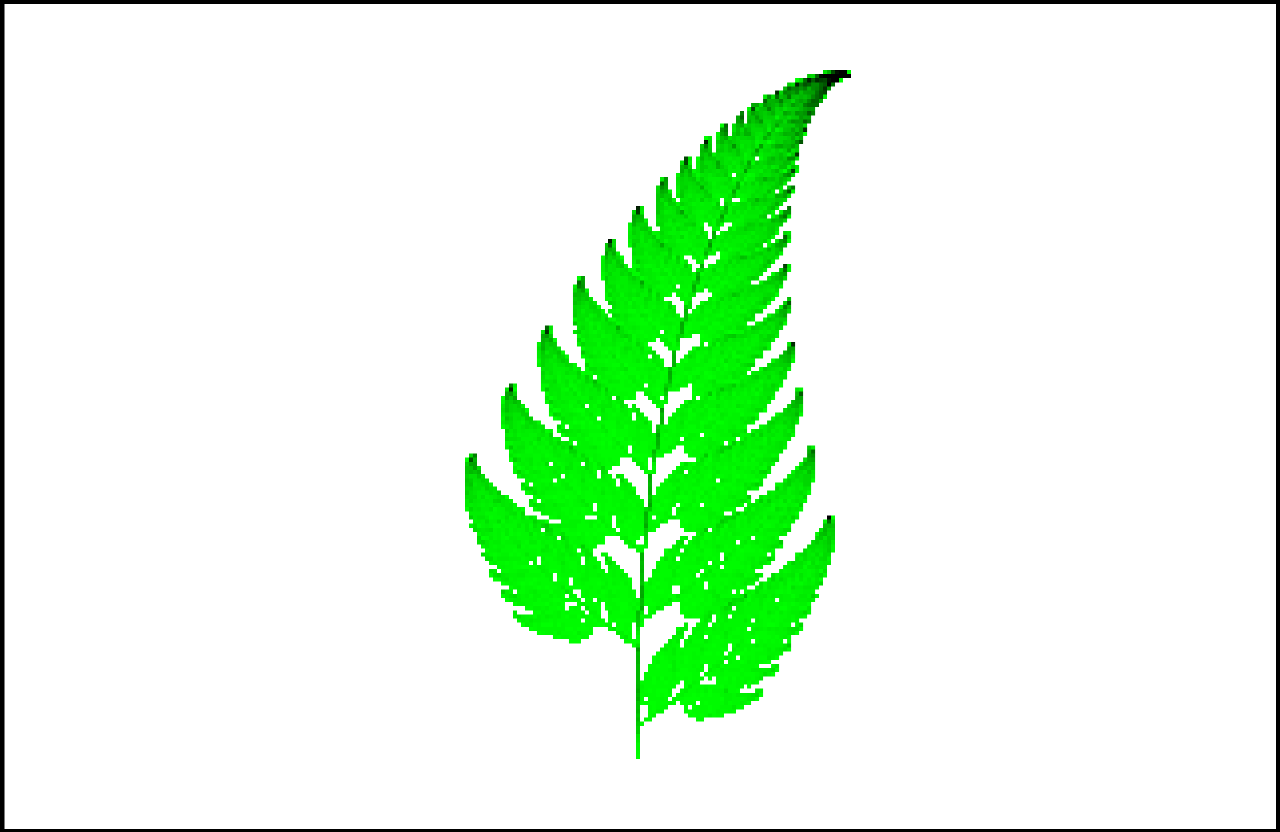
Interpret the stuff inside brackets as a branch.



Lindenmayer 1968



Inductive Leaf



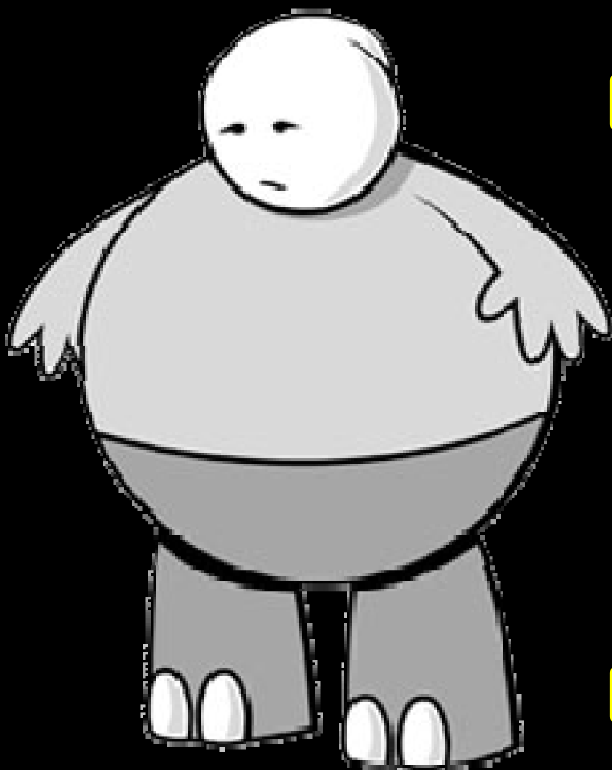
"The Algorithmic Beauty of Plants"

- Start at X
  Sub(X) = F-[[X]+X]+F[+FX]-X
  Sub(F) = FF

- Angle=22.5

Much more stuff at

- http://www.cbc.yale.edu/courseware/swinglsystem.html



Here's What You Need to Know…

Inductive Proof

- Standard Form
- All Previous Form
- Least-Counter Example Form
- Invariant Form

Inductive Definition

- Recurrence Relations
- Fibonacci Numbers
- Guess and Verify