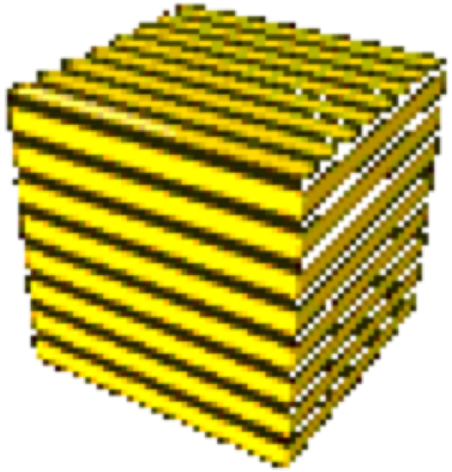
Chaos theory

"CHAOS OFTEN BREEDS LIFE,
WHEN ORDER BREEDS HABIT"
Henry Adams



Chaos theory

- CHAOS is a state of utter confusion or disorder; a total lack of organization or order
- CHAOS is a natural phenomenon involving the physical properties of matter and energy and something extremely sensitive to initial conditions
- Chaos also refers to the question of whether or not it is possible to make good long-term predictions about how a system will act

Chaos theory (history)

1880s	Henri Poincaré (an early proponent of chaos theory)
1898	Jacques Hadamard published an study of the chaotic motion of a free particle
1961	Edward Lorenz - an early pioneer of the theory
1982	Mandelbrot published *The Fractal Geometry of Nature*, which became a classic of chaos theory
1987	Per Bak, Chao Tang and Kurt Wiesenfeld published a paper in *Physical Review Letter* describing for the first time *self-organized criticality*
1997	Ilya Prigogine showed that complex structures could come from simpler ones

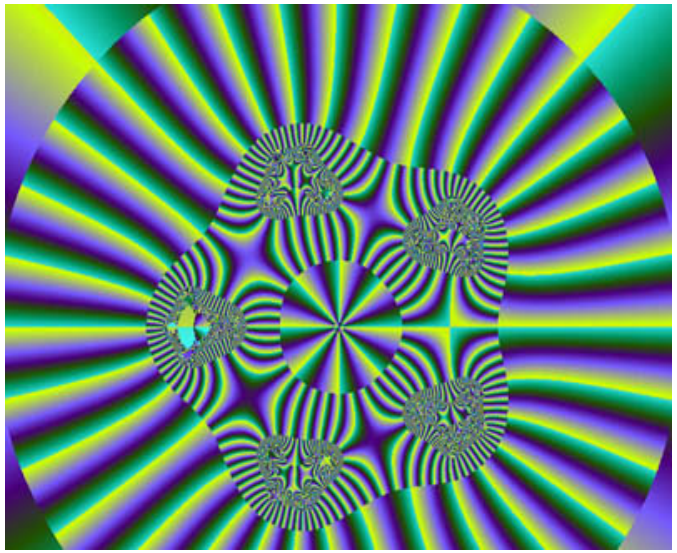
Chaos theory

The theory was summarized by Edward Lorenz as:

Chaos: When the present determines the future, but the approximate present does not approximately determine the future

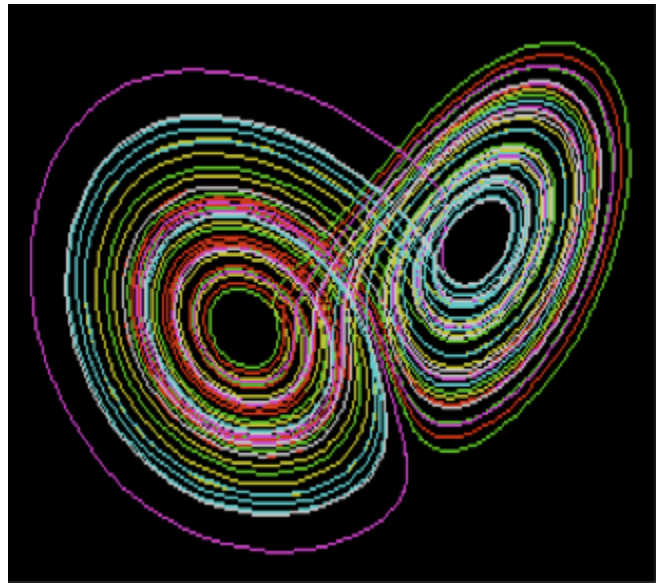
Chaos theory

- A chaotic system can actually develop in a way that appears very smooth and ordered (fractals)
- Applications:
  - geology, mathematics, microbiology, biology, computer science (cryptography), economics, engineering, finance, algorithmic trading, meteorology, philosophy, physics, politics, population dynamics, psychology, and robotics



Chaos theory (complexity)

- Complexity can occur in natural and man-made systems, as well as in social structures and human beings
- Complex dynamical systems may be very large or very small



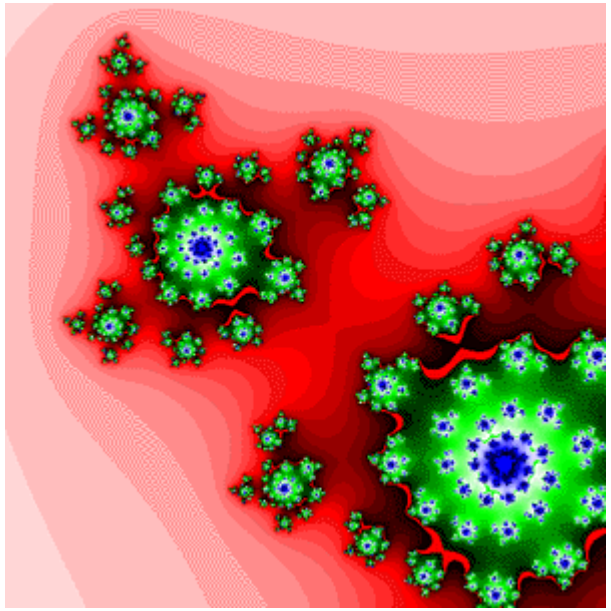
https://commons.wikimedia.org/w/index.php?curid=22398720

Chaos theory

- Computer graphics software is good at creating representations of man-made objects using primitives
  - lines, rectangles, polygons, and curves in 2 D or boxes and surfaces in 3D
- Insufficient when it comes to representing objects found in nature
  - clouds, trees, veins, waves, and a clump of mud
- Many processes in the world can be accurately described using chaos theory and fractal geometry

Chaos theory

- The computer graphics uses techniques to generate images as well as realistic natural looking structures:
  - Chaotic Systems (The classic Mandelbrot)
  - Strange Attractors
  - Newton Raphson
  - Diffusion Limited Aggregation
  - L-Systems (Aristid Lindenmeyer-model plant growth)
  - Iterated Function Systems

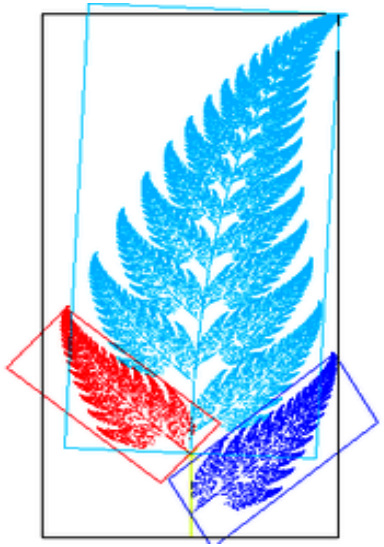


http://paulbourke.net/fractals/interface/

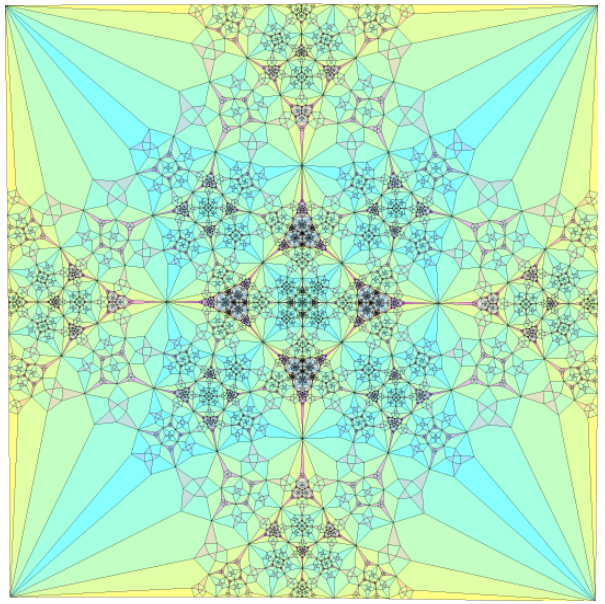
Fractal

A **fractal** is a natural phenomenon or a mathematical **set** that exhibits a repeating pattern that displays at every scale. It is also known as expanding symmetry or evolving symmetry. If the replication is exactly the same at every scale, it is called a **self-similar**

Close-up of a Romanesco broccoli



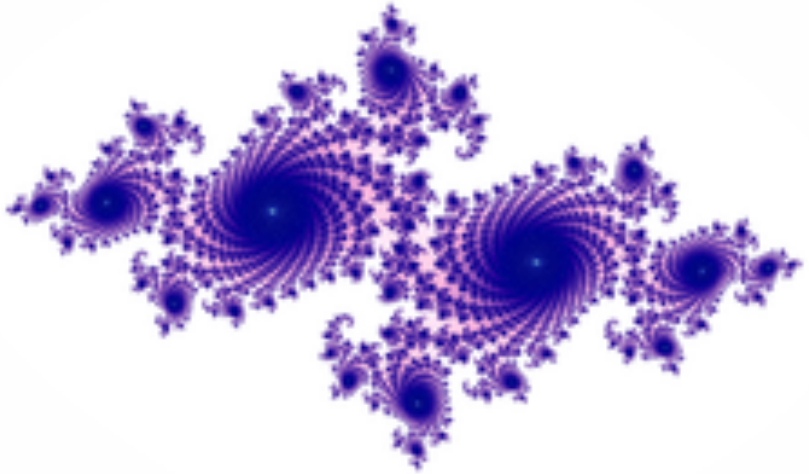
An image of a fern



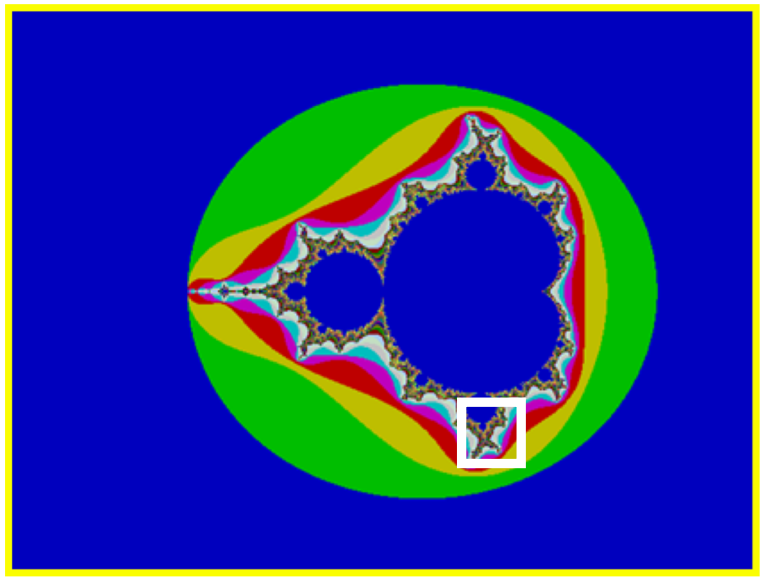
https://commons.wikimedia.org/w/index.php?curid=22398720

Fractal

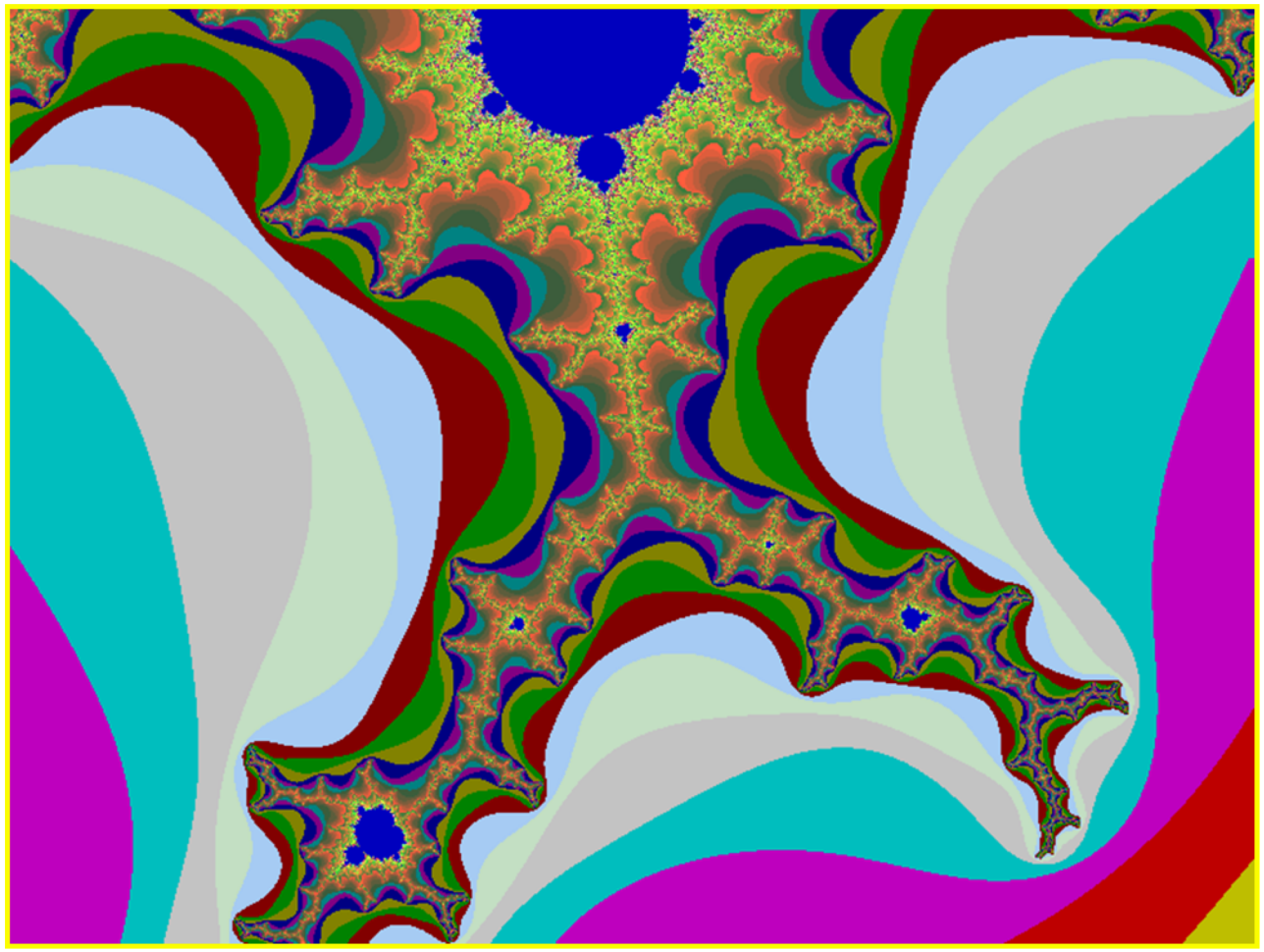
- A fractal is a mathematical object that is both self-similar and chaotic
  - self-similar: As you magnify, you see the object over and over again in its parts
  - chaotic: Fractals are infinitely complex
- Gaston Julia (1918) described the iteration of a rational function



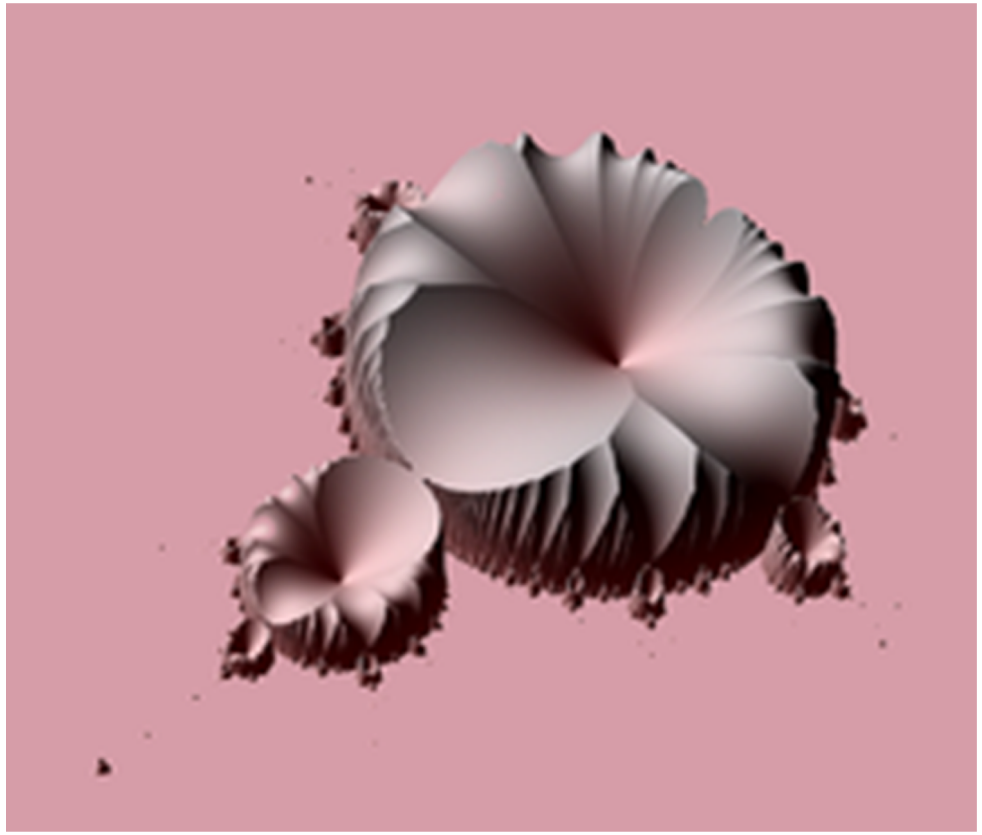
Fractal



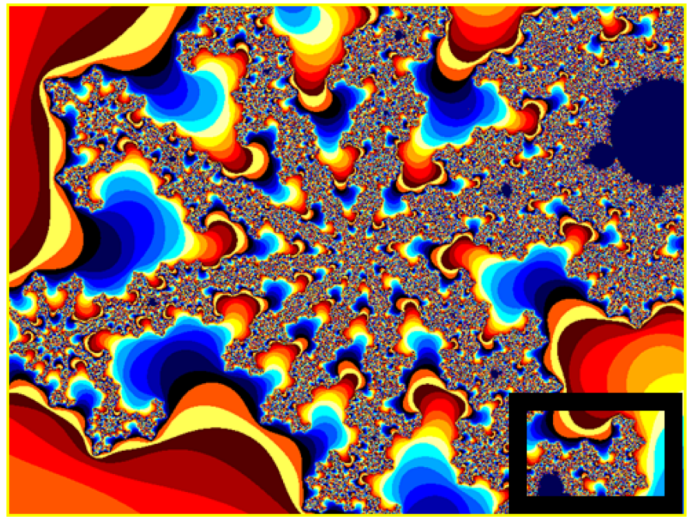
- The most famous of all fractals is the Mandelbrot set

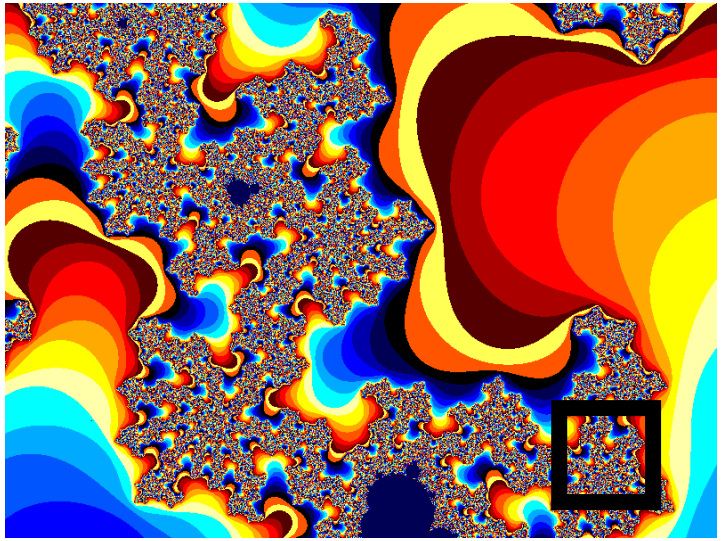


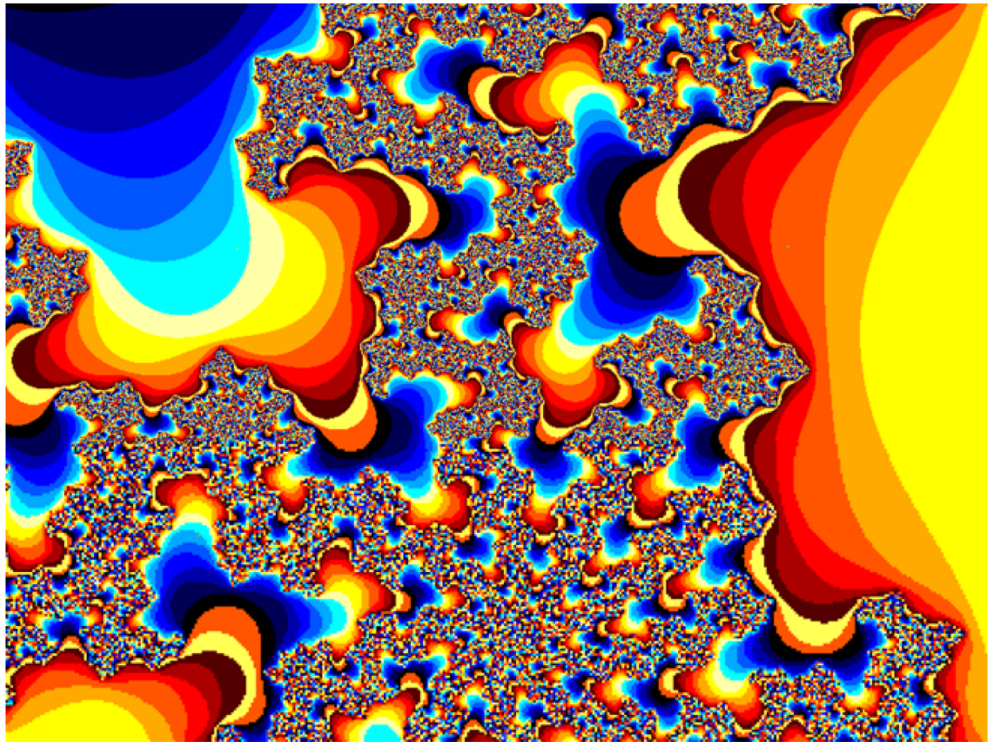
Fractal



Fractal (Self Similarity)



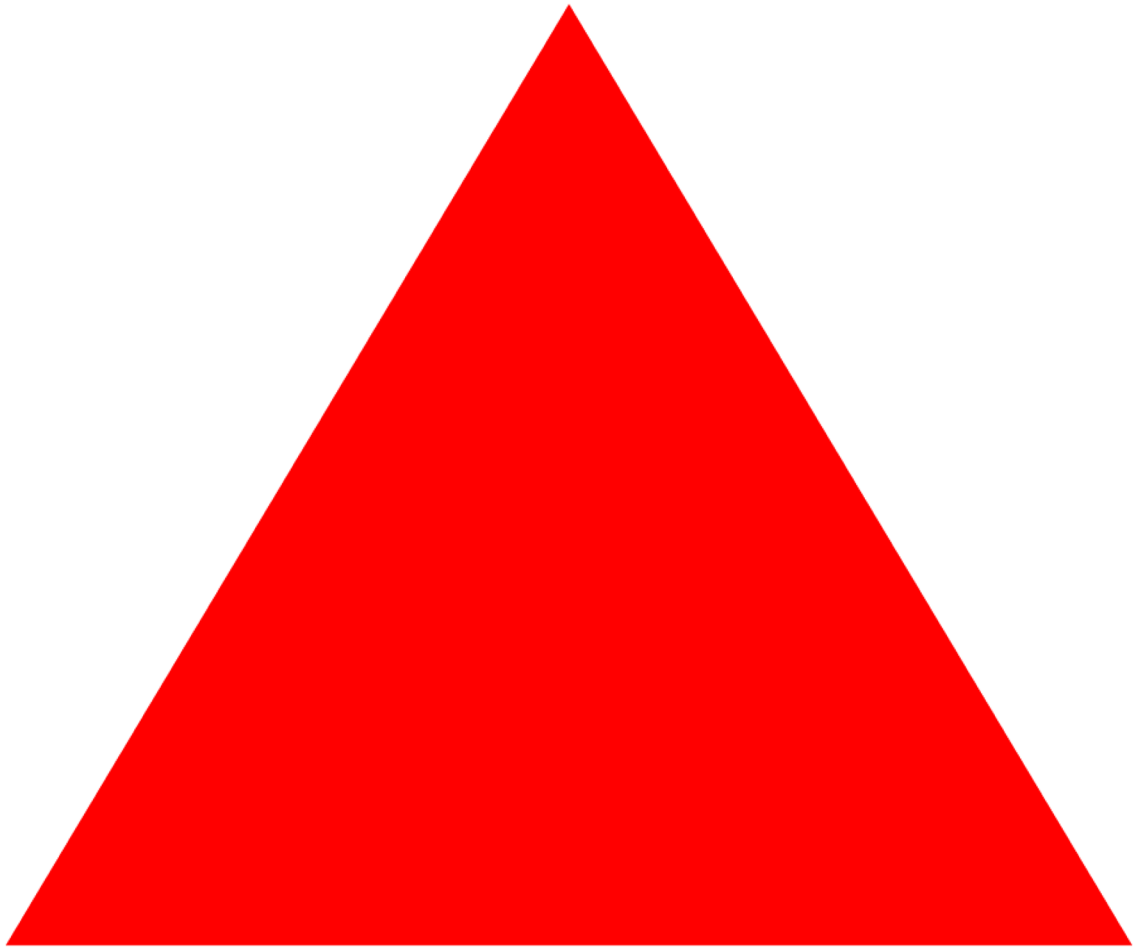




Sierpinski triangle

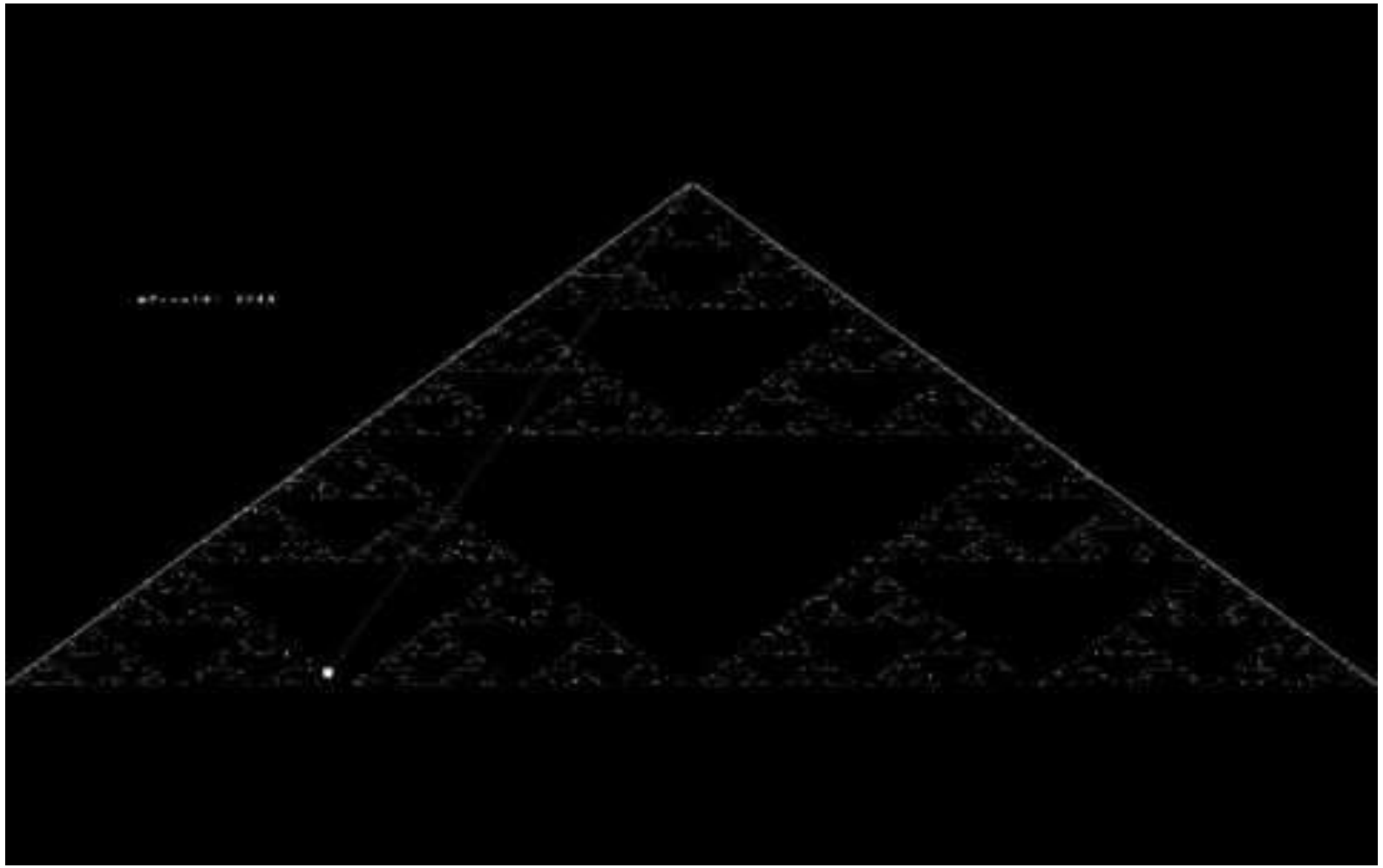
The Sierpinski triangle fractal was first introduced in 1915 by Wacław Sierpiński (1882 –1969). But similar patterns already appeared in the 13th-century in some cathedrals. He was known for outstanding contributions to set theory (research on the axiom of choice and the continuum hypothesis), number theory, theory of functions and topology. He published over 700 papers and 50 books.

Sierpinski triangle



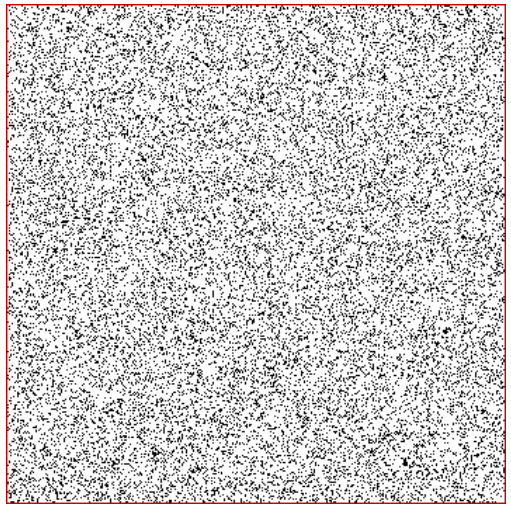
https://commons.wikimedia.org/wiki/File:Animated_construction_of_Sierpinski_Triangle.gif

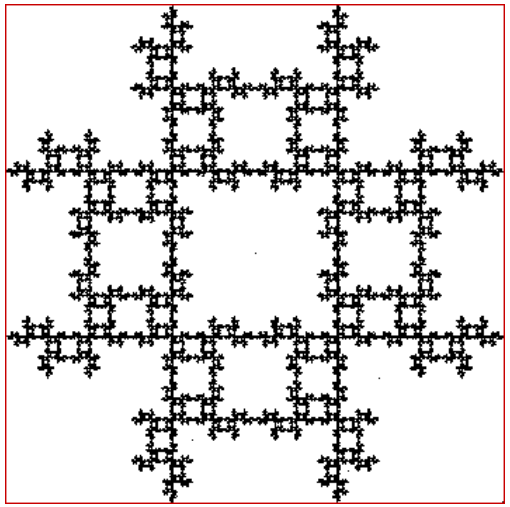
Sierpinski triangle (chaos game)

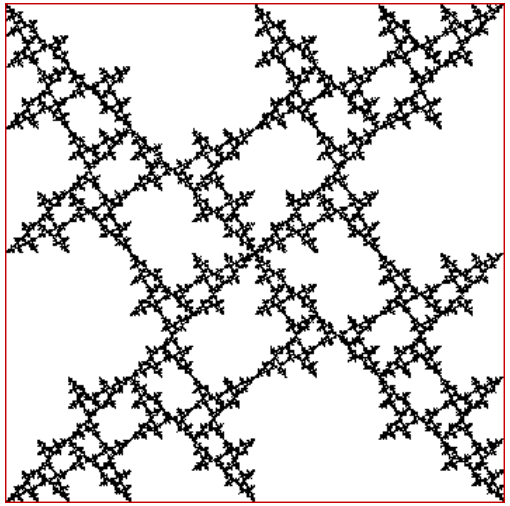


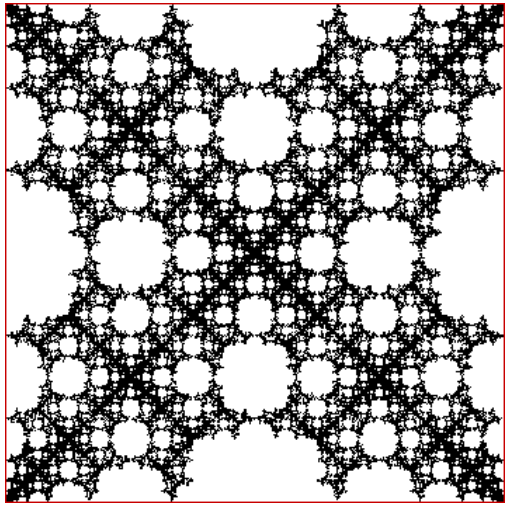
https://youtu.be/droTYSmSGHg

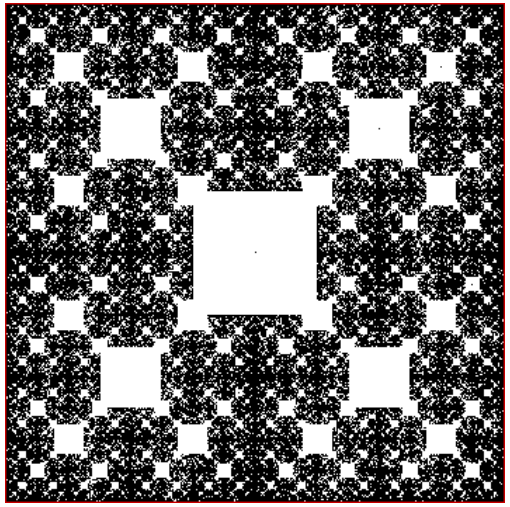
Chaos game





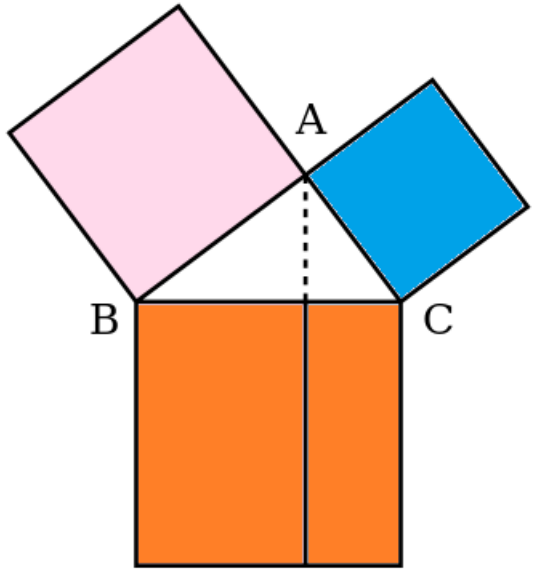




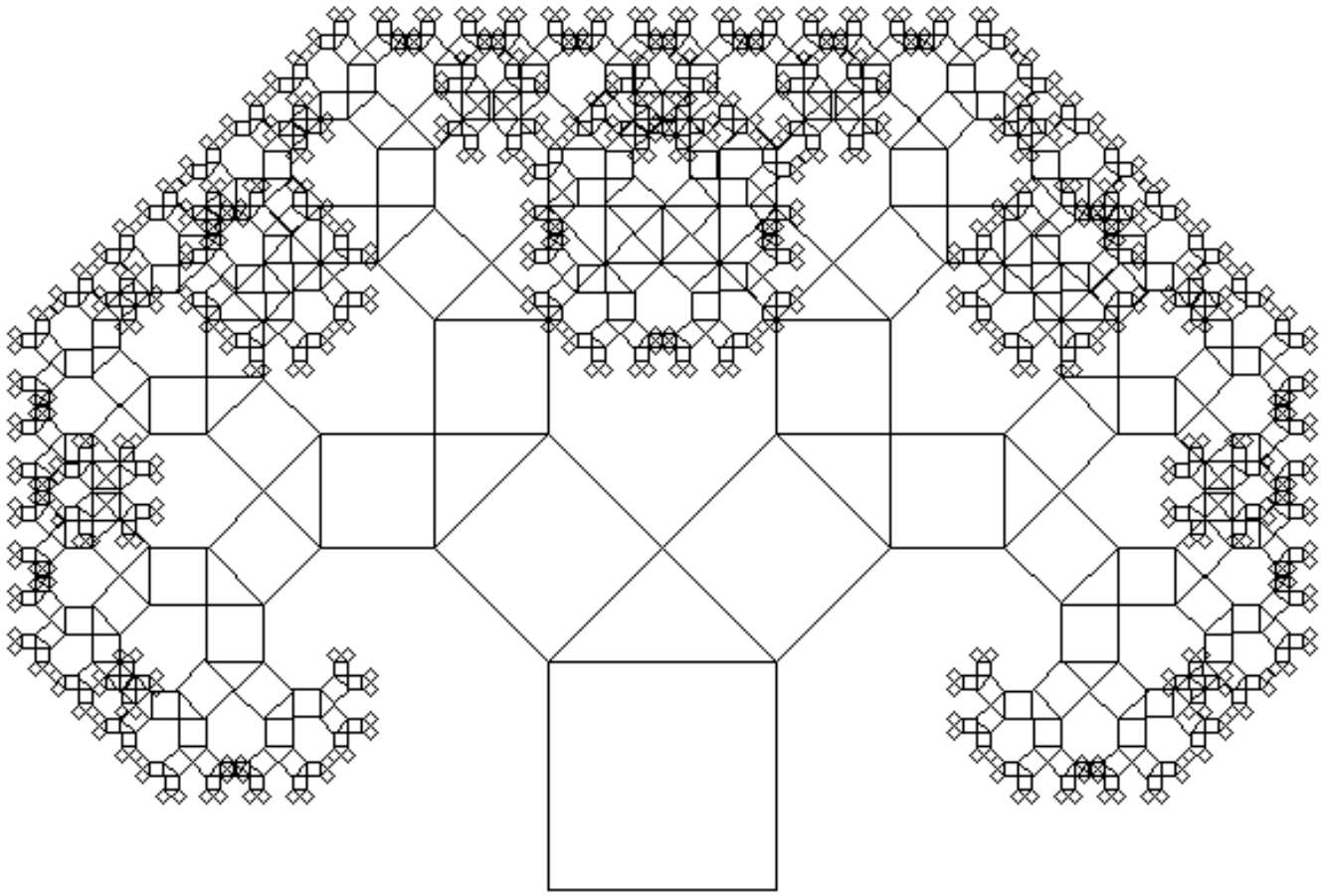


Pythagoras tree

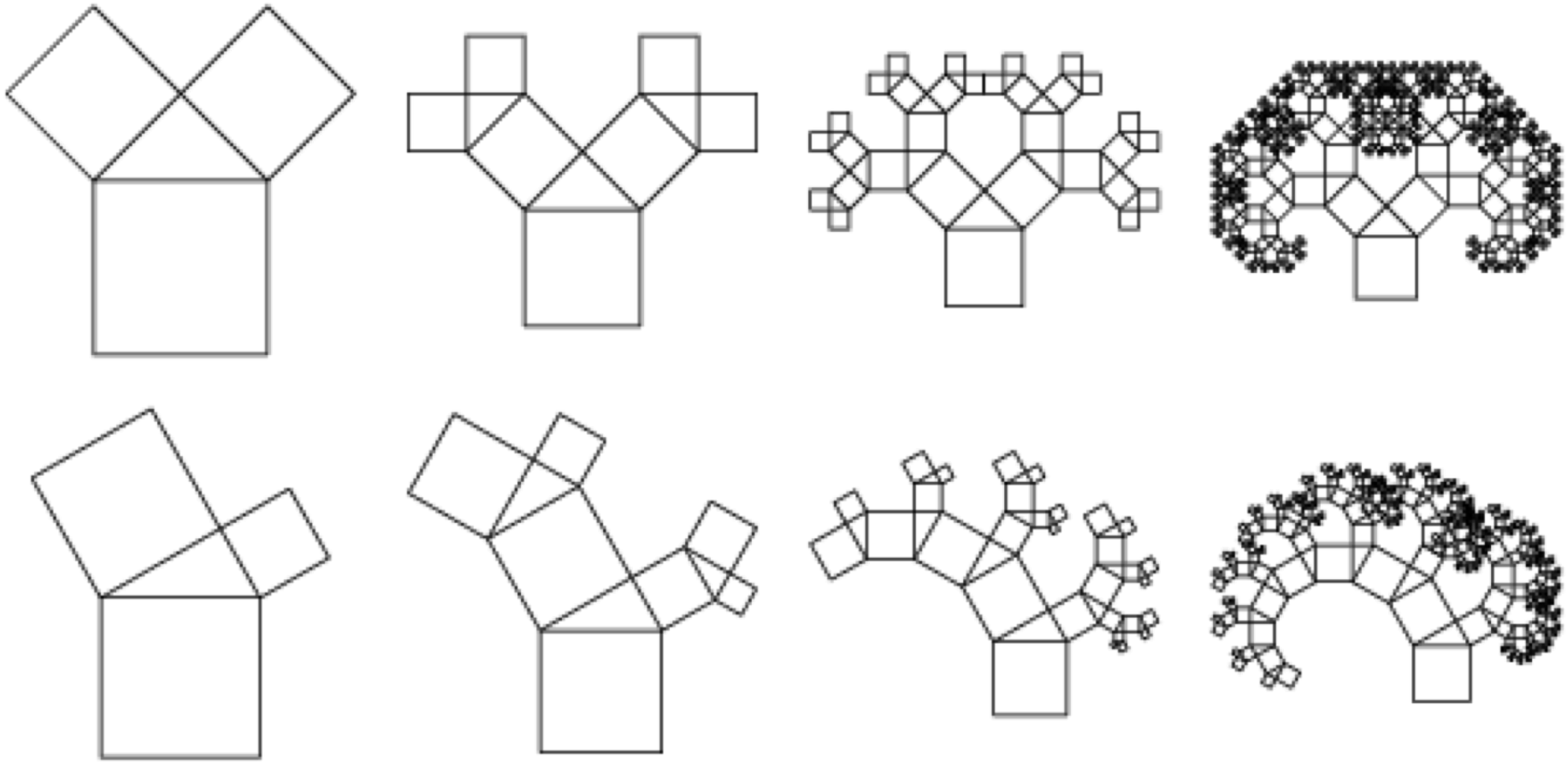
- The Pythagoras tree is a fractal tree constructed from squares
- It is named after Pythagoras because each triple of touching squares encloses a right triangle, in a configuration traditionally used to represent the Pythagorean theorem



Pythagoras tree



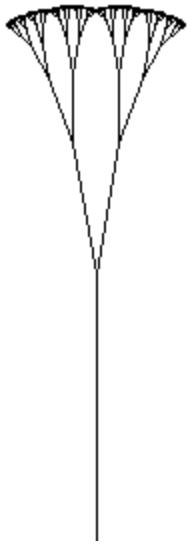
Pythagoras tree



Recursion

- Recursion is perhaps the most powerful programming technique
- The most difficult to grasp
- Solve problems by decomposing the problem into smaller instances of the same problem

Recursion



Lindenmayer **Systems (L-Systems)**

http://fractalfoundation.org/OFC/OFC-2-1.html

Recursion

- In some problems, it may be natural to define the problem in terms of the problem itself.
- Recursion is useful for problems that can be represented by a **simpler version** of the same problem.
- Example: the factorial function

6! = 6 * 5 * 4 * 3 * 2 * 1

We could write:

6! = 6 * 5!

Factorial function

- In general, we can express the factorial function as follows:

```
n! = n * (n-1)!
```

- Is this correct? Well… almost.

  The factorial function is only defined for *positive* integers.

  So we should be a bit more precise:

```
n! = 1                (if n is equal to 1)
n! = n * (n-1)!       (if n is larger than 1)
```

Factorial function

The C++ equivalent of this definition:

```
int fac(int numb){
    if(numb<=1)
        return 1;
    else
        return numb * fac(numb-1);
}
```

Factorial function

- Assume the number typed is 3, that is, numb=3.

```
3 <= 1 ?                    No.
fac(3) = 3 * fac(2)
   fac(2) :
      2 <= 1 ?              No.
      fac(2) = 2 * fac(1)
          fac(1) :
             1 <= 1 ?       Yes.
             return 1
       fac(2) = 2 * 1 = 2
       return fac(2)
 fac(3) = 3 * 2 = 6
 return fac(3)
 fac(3)  has the value 6
```

```
int fac(int numb){
  if(numb<=1)
      return 1;
  else
      return numb * fac(numb-1);
}
```

Factorial function

- For certain problems (such as the factorial function), a recursive solution often leads to short and elegant code. Compare the recursive solution with the iterative solution:

Recursive solution

```
int fac(int numb){
  if(numb<=1)
    return 1;
  else
    return numb*fac(numb-1);
}
```

Iterative solution

```
int fac(int numb){
      int product=1;
      while(numb>1){
            product *= numb;
            numb--;
      }
return product
}
```

Recursion

- Price of recursion:
  - calling a function consumes more time and memory than adjusting a loop counter
  - high performance applications (graphic action games, simulations of nuclear explosions) hardly ever use recursion
- Recursion is one way to decompose a task into smaller subtasks
- At least one of the subtasks is a smaller example of the same task
- The smallest example of the same task has a non-recursive solution

Tail Recursion

- A recursive function is tail recursive when recursive call is the last thing executed by the function
- **Example**: function `print()` is tail recursive

```
void print(int n)
{
    if (n < 0)  return;
    cout << " " << n;
    // The last executed statement is recursive call
    print(n-1);
}
```

#include guard

- **#include guard**, sometimes called a **macro guard**, is a particular construct used to avoid the problem of *double inclusion*

File "grandfather.h“

```
struct foo {
      int member;
};
```

File "father.h“

```
#include "grandfather.h"
```

File "child.c"

```
#include "grandfather.h"
#include "father.h"
```


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

#include guard

File "grandfather.h“

```
#ifndef GRANDFATHER_H
#define GRANDFATHER_H
struct foo
{
      int member;
};
#endif /* GRANDFATHER_H */
```

File "father.h“

```
#include "grandfather.h"
```

File "child.c"

```
#include "grandfather.h"
#include "father.h"
```


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