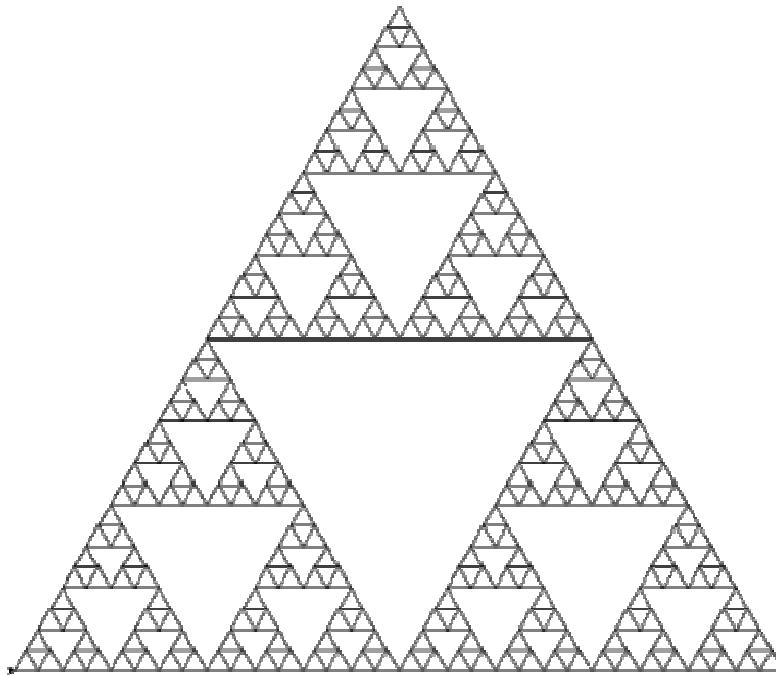


Moving from Beaver to Olympiad



Ries Kock

National Institute for Curriculum Development
in The Netherlands



Beaver and Olympiad

- Beaver: to find gifted children
- Olympiad: to develop gifted children



Bebras in The Netherlands

- Bebras
 - Age groups II, III, IV
 - Covers secondary education
 - First round in November
 - Second round end of January
 - The best 50 of each group are invited
 - At a University
 - Same kind of contest
 - Presentations about the tasks for parents / teachers
 - Numbers are still increasing:
 - 2005: 250
 - 2010: 10200



Olympiad in Informatics in The Netherlands

- Age group III, IV
 - First round: September to January
 - Second round: March
 - Schooling for the third round: April / May
 - Third round: June
-
- First round: 500
 - Decreasing



Foundation of Olympiad in Informatics

Organizes three contests:

- Bebras
- Olympiad in Informatics
- CodeCup (open for the world)



How to connect to the success of Bebras?

Remember:

- Bebras: two times 40 minutes
- First round olympiad: at least 40 hours



Some ideas

- Start with Bebras tasks
- Connect programming to it
- Using just a few elements of the Python language
- Interesting results
- It should be about problem solving
- Several modules of about 10? hours



Why Python?

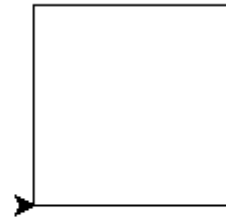
- Free
- Portable
- Analyses show: suitable for education
- Turtle graphics available
- A USACO like environment possible?

Module 1

1. Introduction
2. Install Python
3. Direct mode:

```
>>> for step in range(4):  
    t.forward(100)  
    t.left(90)
```

```
>>>
```

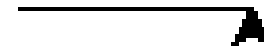


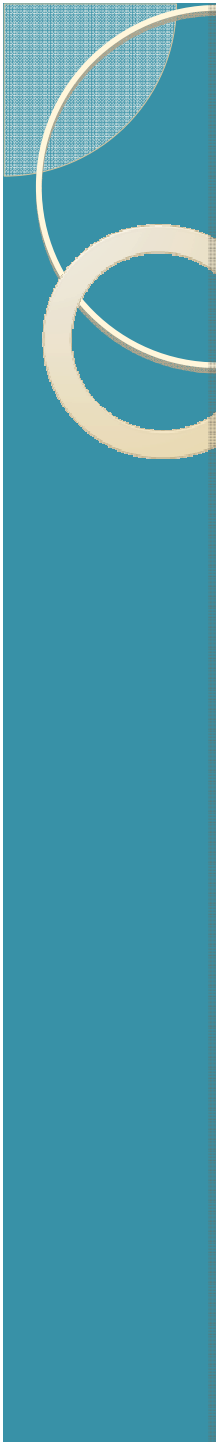
- First Bebras task

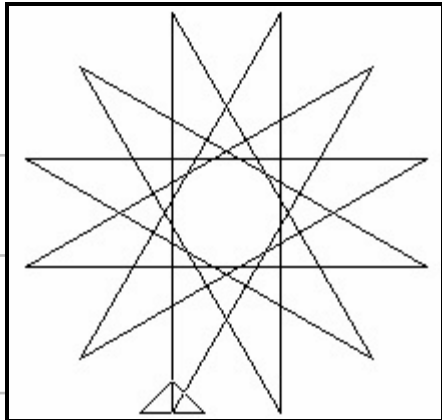
Use turtle module

Direct mode:

```
>>> import turtle  
>>> t = turtle.Pen()  
>>> t.forward(60)  
>>> t.left(90)  
>>>
```





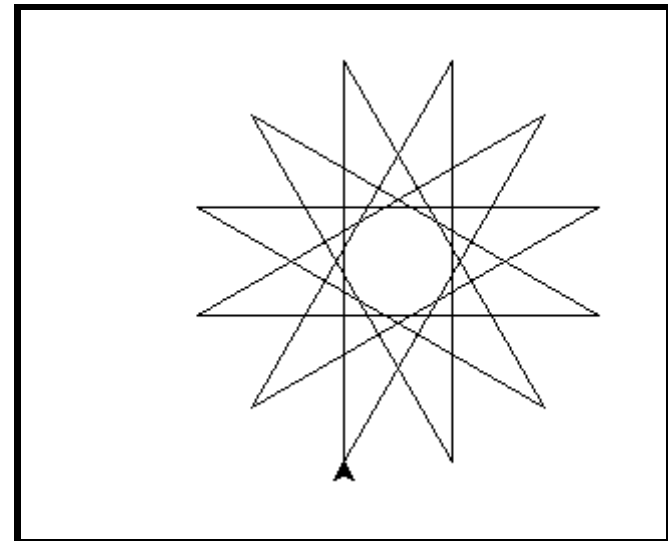
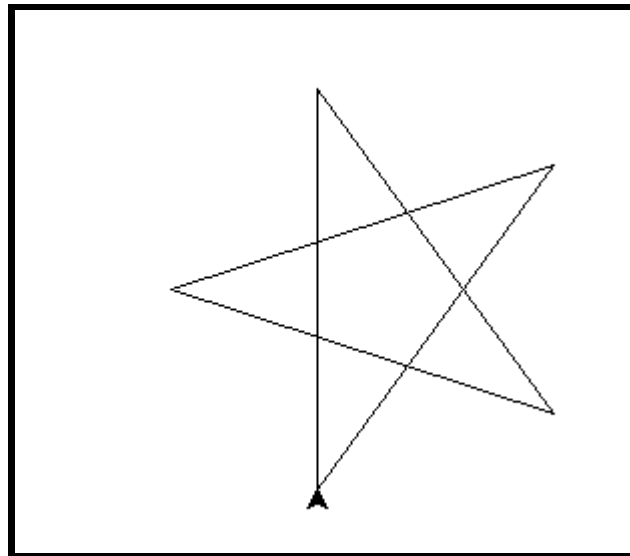
Bebras star Turtle Bebras can do the following	
<code>forward n</code>	
<code>right α</code>	
<code>left α</code>	
<code>repeat n [forward n right α]</code>	
<i>Given the figure. Which of the following lines has this result.</i>	
A	<code>repeat 12 [forward 200 right 150]</code>
B	<code>repeat 12 [forward 200 right 30]</code>
C	<code>repeat 10 [forward 200 right 144]</code>
D	<code>repeat 10 [forward 200 right 36]</code>

Task

Interactive task?

What is the relation between the number1 and number2, when the result is a star?

```
for x in range(number1):  
    t.forward(100)  
    t.right(number2)
```



- 
- Star
 - not a star

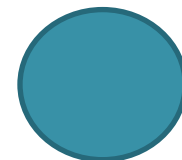


Definition square

```
import turtle
t=turtle.Pen()
turtle.mode ("logo")

def square (side):
    for step in range (4):
        t.forward(side)
        t.right(90)

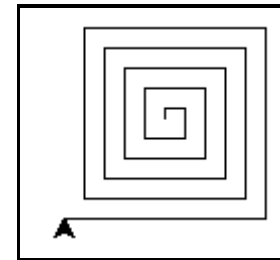
square (100)
square (200)
turtle.exitonclick()
```



A function who calls a function

```
def spiral(side):  
    if side < 200:  
        t.forward(side)  
        t.right(90)  
        spiral(side + 5)
```

```
spiral(5)
```

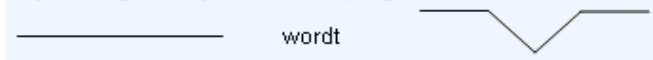


spiral

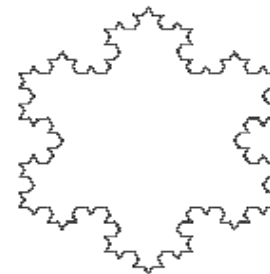
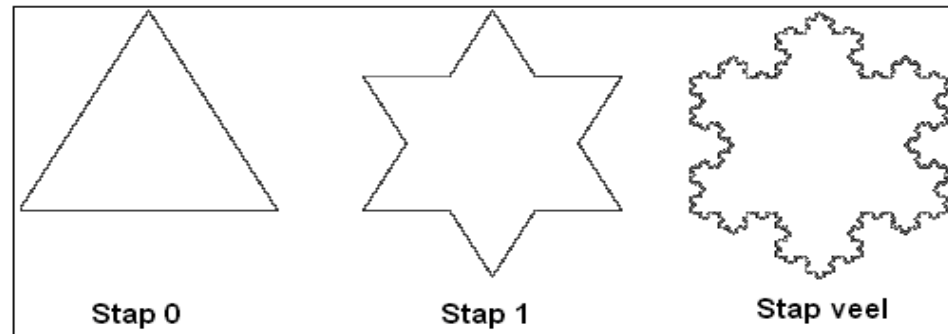
Another Bebras task

Fractal van Koch

Een fractal is een kunstig beeld met een oneindig (nou ja, heel vaak) herhaald patroon. Een voorbeeld van een fractal is de sneeuwvlok van Koch. Om de sneeuwvlok te maken begin je met een driehoek (stap 0). Vervolgens vervang je elk lijnstuk volgens het volgende model:
(Bij een figuur wijst de uitstulping naar de buitenkant.)



Hieronder zie je een aantal stappen van de sneeuwvlok van Koch



Na hoeveel stappen krijg je de volgende sneeuwvlok?

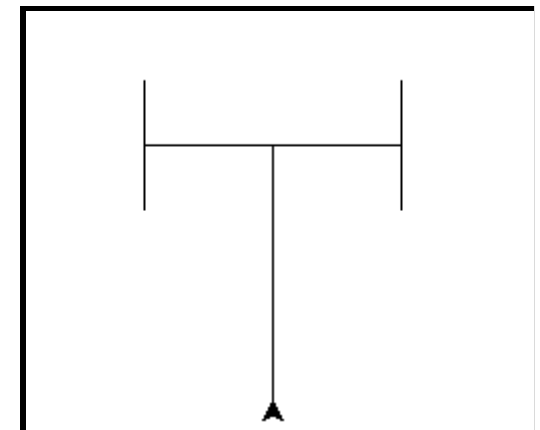
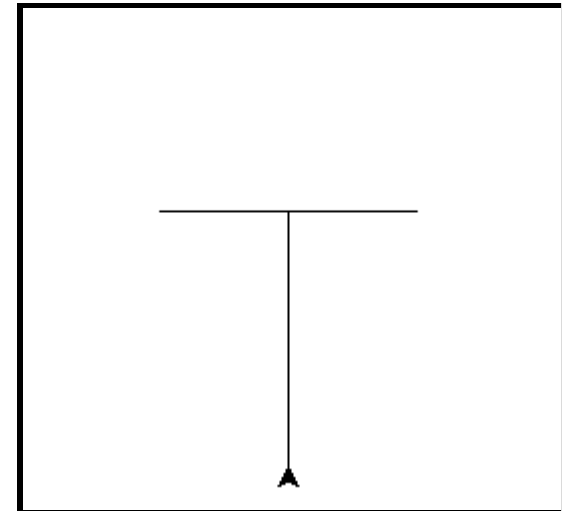
Hfractal

```
def Hfractal1 (side):  
    t.forward(side)      # go forward  
    t.left(90)           # look left  
    t.right(180)         # look right  
    t.left(90)           # look straight ahead  
    t.back(side)         # go backward
```

```
def Hfractal2(side):  
    t.forward(side)      # go forward  
    t.left(90)           # look left  
    Hfractal1(side/2)  
    t.right(180)         # look right  
    Hfractal1(side/2)  
    t.left(90)           # look straight ahead  
    t.back(side)         # go backward
```

```
Hfractal2(128)
```

Hfractal2(128)



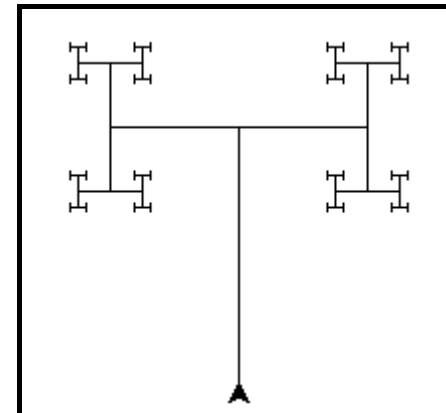
Write Hfractal3

Hfractal (shorter)

```
def Hfractal(side):  
    if side > 4:  
        t.forward(side)    # go forward  
        t.left(90)         # look left  
        Hfractal(side/2)  
        t.right(180)       # look right  
        Hfractal(side/2)  
        t.left(90)         # look straight ahead  
        t.back(side)       # go backward
```

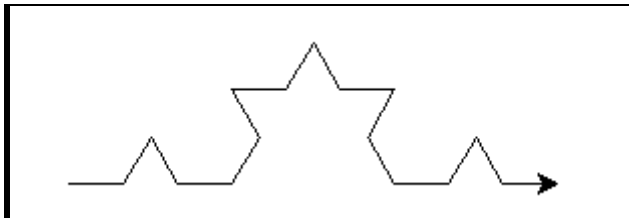
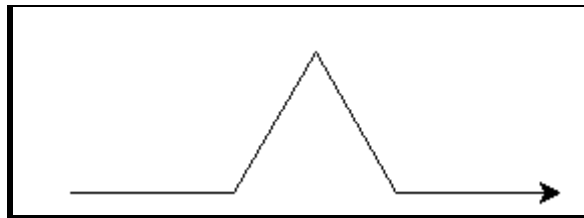
```
Hfractal(128)
```

Hfractal





Koch fractal



```
import turtle
t=turtle.Pen()

def koch1(zijde):
    t.fd(zijde)

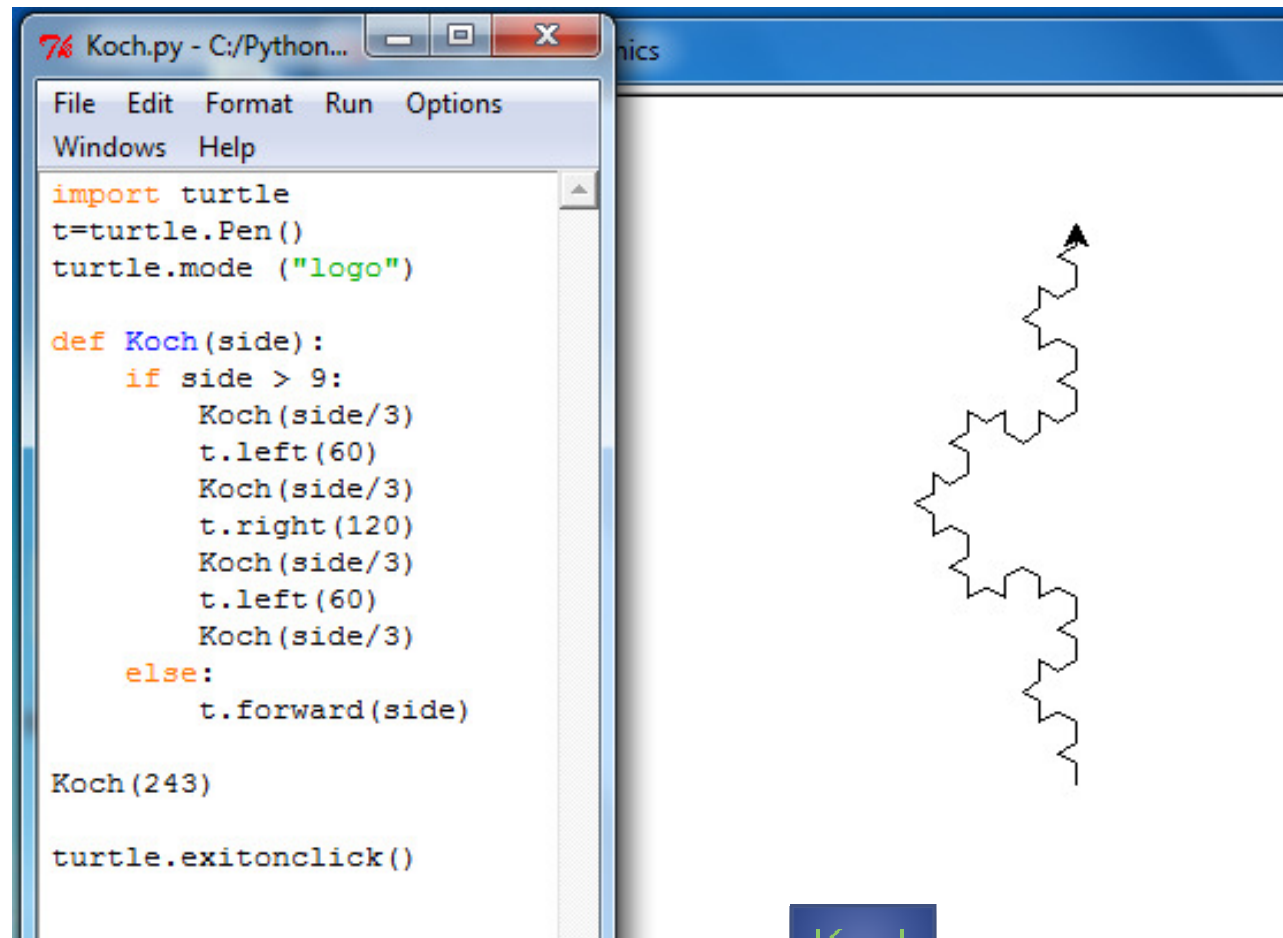
def koch2(zijde):
    koch1(zijde/3)
    t.left(60)
    koch1(zijde/3)
    t.right(120)
    koch1(zijde/3)
    t.left(60)
    koch1(zijde/3)

def koch3(zijde):
    koch2(zijde/3)
    t.left(60)
    koch2(zijde/3)
    t.right(120)
    koch2(zijde/3)
    t.left(60)
    koch2(zijde/3)

koch3(243)

turtle.exitonclick()
```

Koch shorter



```
76 Koch.py - C:/Python...
File Edit Format Run Options
Windows Help

import turtle
t=turtle.Pen()
turtle.mode ("logo")

def Koch(side):
    if side > 9:
        Koch(side/3)
        t.left(60)
        Koch(side/3)
        t.right(120)
        Koch(side/3)
        t.left(60)
        Koch(side/3)
    else:
        t.forward(side)

Koch(243)

turtle.exitonclick()
```

The image shows a screenshot of a Python IDE window titled "76 Koch.py - C:/Python...". The window contains a menu bar with "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The code editor displays the following Python code for drawing a Koch curve:

```
import turtle
t=turtle.Pen()
turtle.mode ("logo")

def Koch(side):
    if side > 9:
        Koch(side/3)
        t.left(60)
        Koch(side/3)
        t.right(120)
        Koch(side/3)
        t.left(60)
        Koch(side/3)
    else:
        t.forward(side)

Koch(243)

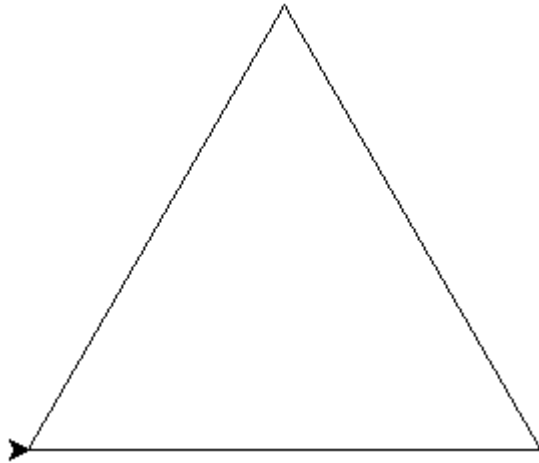
turtle.exitonclick()
```

To the right of the code editor, a black fractal curve is drawn on a white background. The curve is a Koch curve, which is a self-similar fractal. It starts at the bottom left and moves upwards and to the right, ending with an arrowhead. The curve is composed of many small segments, each of which is a scaled-down version of the whole curve.

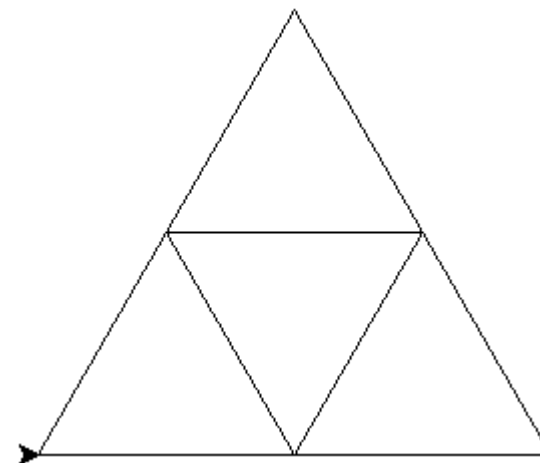
Koch

Sierpinski

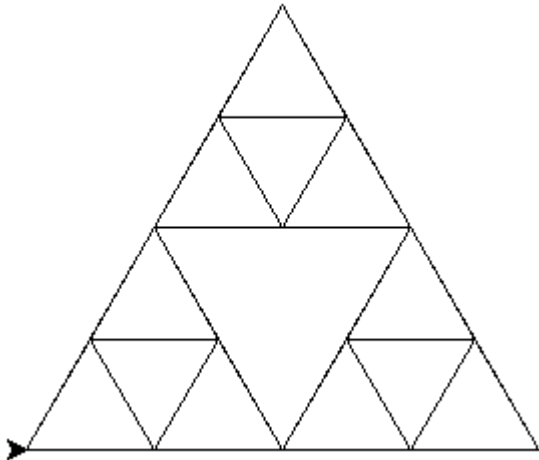
Sierpinski1



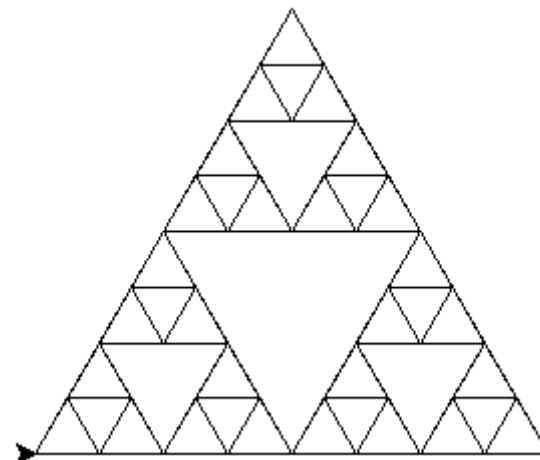
Sierpinski2



Sierpinski3

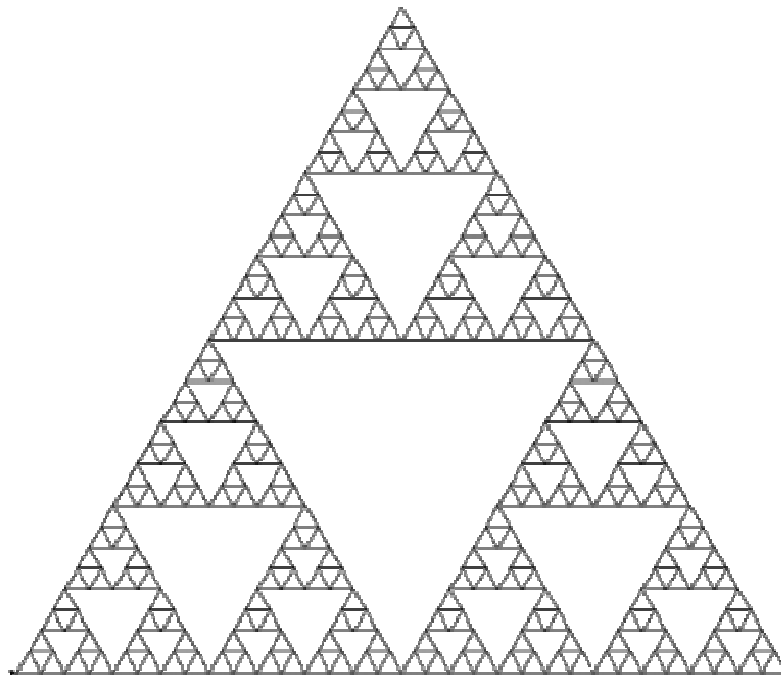


Sierpinski4

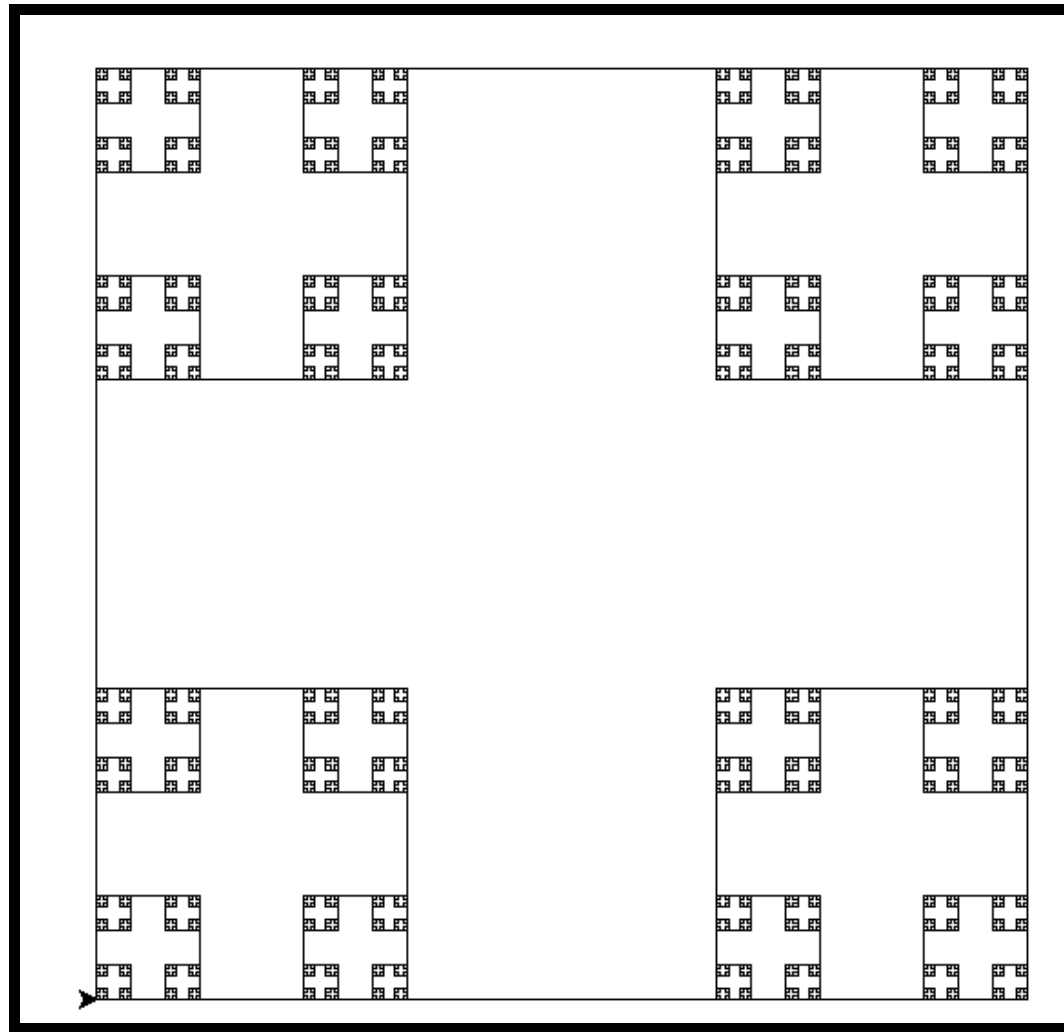


Sierpinski

Sierpinski



Task: Swisspinski





How to implement this module?

- Make it available via the Internet?
- Use it as a module to study for the second round?
- Make a USACO-like environment
- Make an interactive paper?
- What about the second module?
- Thank you for your attention.