

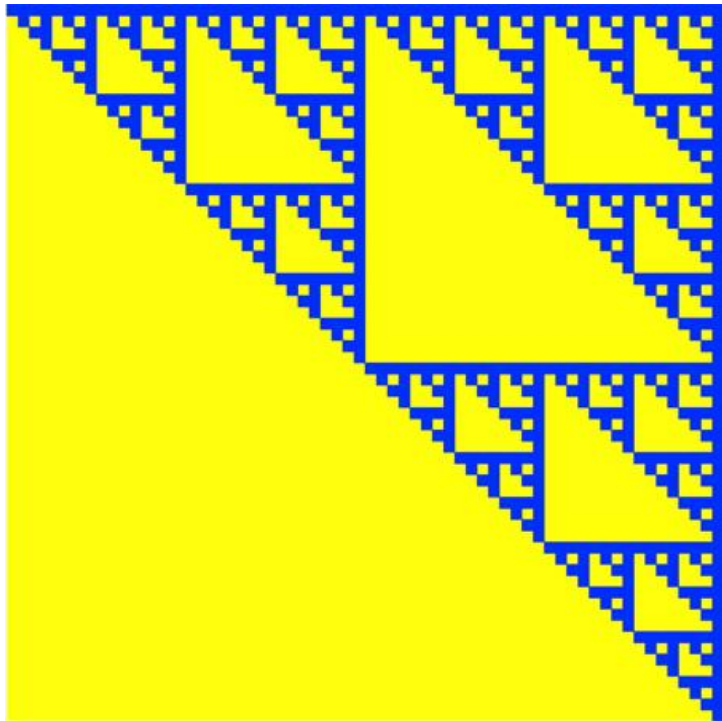


# Algoritmika

6. szeminárium

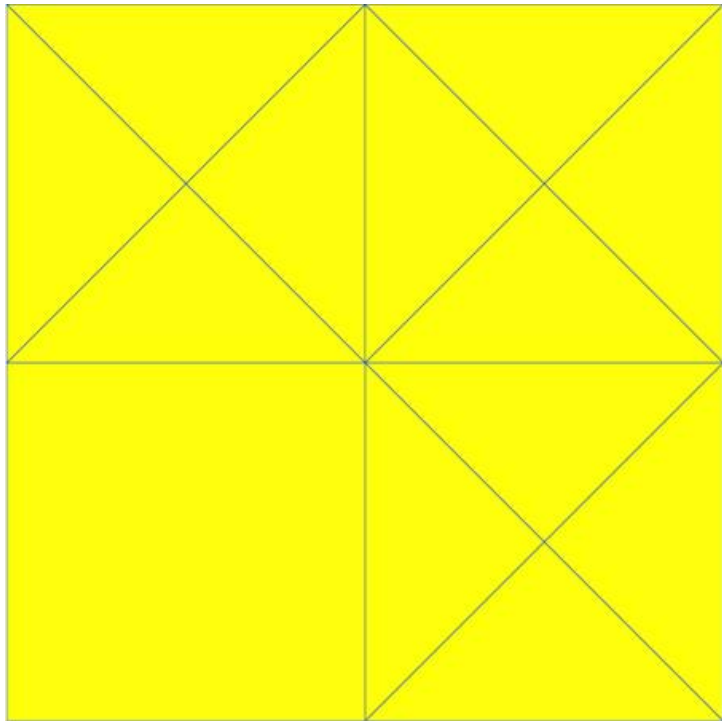


# Sierpiński háromszög



- Wacław Sierpiński lengyel matematikus által megtalált fraktál
- egy szabályos háromszögből elhagyjuk az oldalfelező pontok összekötésével nyert belső háromszöget
- majd az így maradt három háromszögre rekurzívan alkalmazzuk ugyanezt az eljárást.

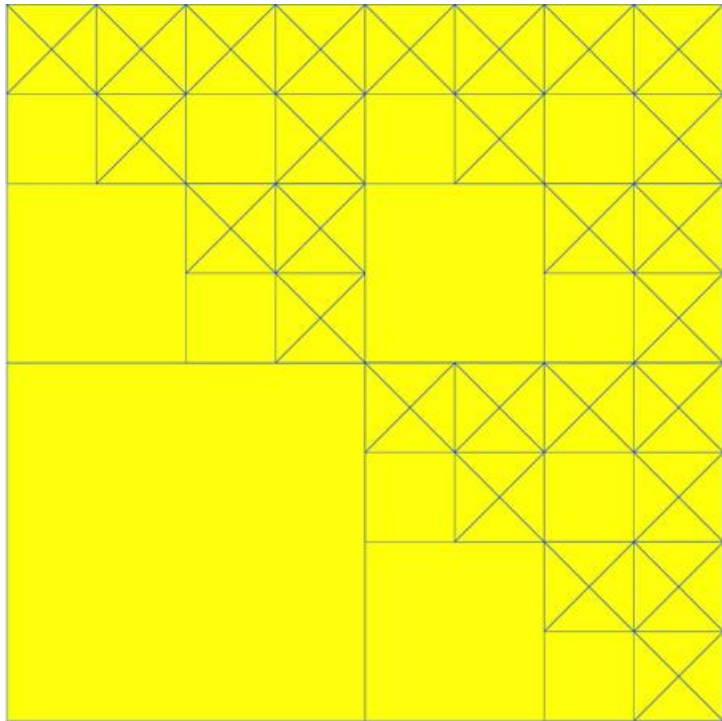
# Sierpiński háromszög



- Egy négyzetet 4 egyenlő részre osztunk
- A bal alsó sarok “üresen” marad
- Az maradék 3 részt tovább osztjuk

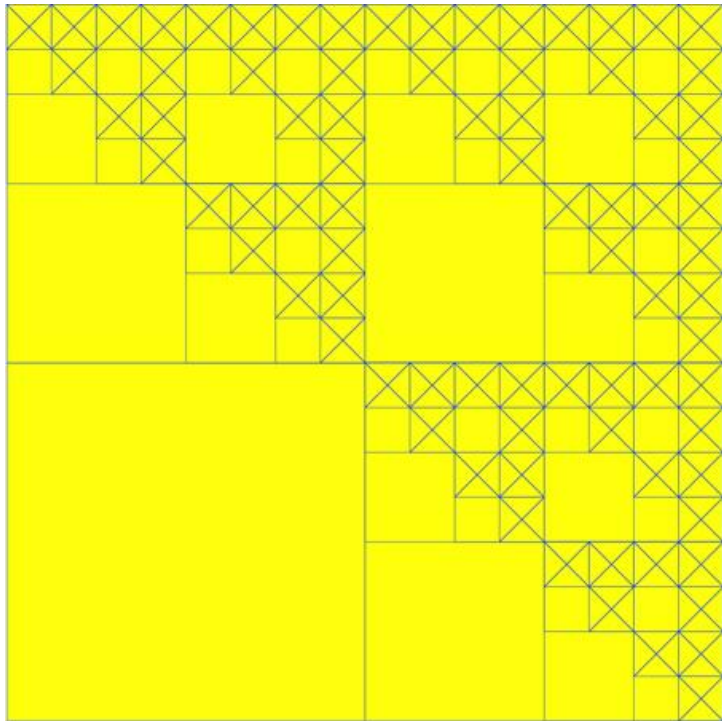


# Sierpiński háromszög



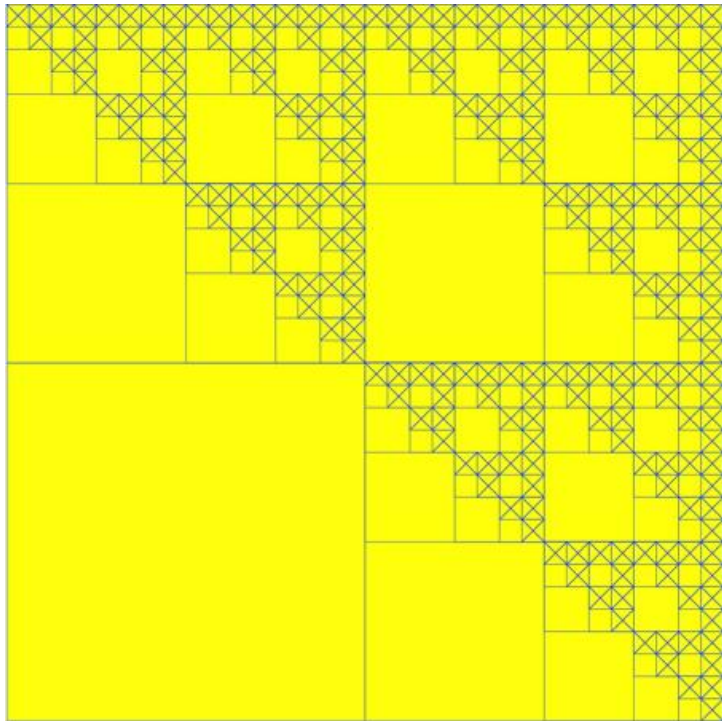
- Egy négyzetet 4 egyenlő részre osztunk
- A bal alsó sarok “üresen” marad
- Az maradék 3 részt tovább osztjuk

# Sierpiński háromszög



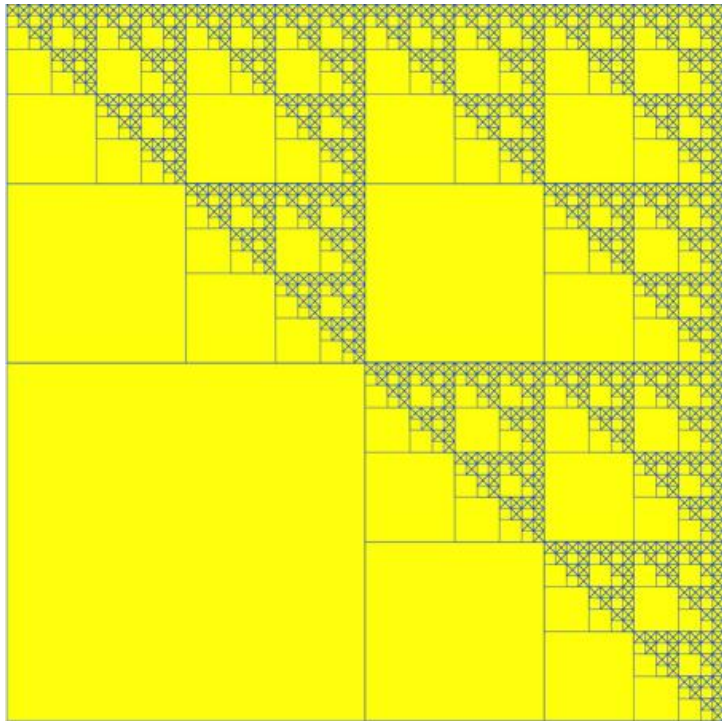
- Egy négyzetet 4 egyenlő részre osztunk
- A bal alsó sarok “üresen” marad
- Az maradék 3 részt tovább osztjuk

# Sierpiński háromszög



- Egy négyzetet 4 egyenlő részre osztunk
- A bal alsó sarok “üresen” marad
- Az maradék 3 részt tovább osztjuk

# Sierpiński háromszög



- Egy négyzetet 4 egyenlő részre osztunk
- A bal alsó sarok “üresen” marad
- Az maradék 3 részt tovább osztjuk
- Addig ismételjük amíg a négyzetecskék “elég” kicsik lesznek



# Sierpiński háromszög

Az algoritmus:

- Amennyiben elég kicsi a négyzet
  - színezd ki
- Különben:
  - Oszd 4 részre a négyzetet
  - Rajzolj Sierpiński háromszöget a bal felső részbe
  - Rajzolj Sierpiński háromszöget a jobb felső részbe
  - Rajzolj Sierpiński háromszöget a jobb alsó részbe

# Rekurzió

**Egyszerűsíts és delegálj!**

- ha egy feladat elég egyszerű: oldd meg



**Alap eset**

- ha nem, csökkentsd le a feladatot egy vagy több  
kisebb méretű, ugyanolyan feladatra



**Rekurzív hívás**

# Orosz szorzás

**Adott  $a$ ,  $b$ . Számoljuk ki  $a*b$ -t rekurzívan.  
(Használjuk az orosz paraszt módszerét)**

**Példa:     $a = 85$ ,  $b=18$   
           $a*b = 1530$**

# Orosz szorzás

**Algoritmus** OroszSzorzasRek (a, b)

Ha  $a=0$  akkor

Vissza: 0

Különben:

Ha  $a \bmod 2 = 1$  akkor

Vissza:  $b + \text{OroszSzorzasRek}(a/2, b*2)$

Különben

Vissza:  $\text{OroszSzorzasRek}(a/2, b*2)$

Vége (Ha)

Vége (Ha)

Vége (Algoritmus)

# Fibonacci

**Generáljuk az  $n$ . Fibonacci számot.**

**Algoritmus** Fibonacci( $n$ )

Ha  $n < 2$  akkor

visszatérít  $n$

Különben

visszatérít Fibonacci( $n - 1$ )+Fibonacci( $n - 2$ )

Vége (Ha)

Vége (Algoritmus)

**Kérdés: Hány rekurzív hívást végez az algoritmus? (pl.  $n = 6$ -ra)**

# Fibonacci

**A fenti megoldás exponenciális futási időhöz vezet, egy részfeladatot többször is megoldunk.**

**Lehetséges megoldás:**

- **a kiszámolt értékeket megőrizzük egy sorozatban**
- **a minden új elem kiszámításánál felhasználjuk a korábban már kiszámolt és elmentett elemeket.**

# Fibonacci

**Algoritmus** Fibonacci2(n, F)

Ha  $n > 2$  akkor

Fibonacci2(n-1, F)

$F[n] \leftarrow F[n-1] + F[n-2]$

Különben

$F[1] \leftarrow 1$

$F[2] \leftarrow 1$

Vége (Ha)

Vége (Algoritmus)

# Inko

**Határozzuk meg  $n$  szám Inko-ját.**

**Megoldás:**

- **írunk egy rekurzív függvényt amely meghatározza két szám Inko-ját**
- **meghatározzuk rekurzívan  $n$  szám Inko-ját:**
  - Egy elemű sorozatnak az Inko-ja maga az elem
  - $n$  elem esetén kiszámoljuk  $n-1$  elem Inko-ját, majd meghatározzuk az  $n$ -dik elem és a kiszámolt Inko Inko-ját.