

Recursividade

SSC0301

Prof. Márcio Delamaro

Recursividade

- Vimos que uma função pode chamar outra
- Uma função pode chama ela mesma
- Faz sentido?

Fatorial

- Qual o valor de $n!$?
 - a multiplicação de todos os inteiros de 1 até n
- `def fatorial(n):`
 - `f = 1`
 - `for k in range(1,n+1):`
 - `f *= k`
 - `return f`

Ou...

Fatorial

- Qual o valor de $n!$?
 - a multiplicação de todos os inteiros de 1 até n
 - $n * (n-1)!$. Além disso, $0! = 1$
- $4! = 4 * 3! = 4 * 3 * 2! = 4 * 3 * 2 * 1! = 4 * 3 * 2 * 1 * 0! = 4 * 3 * 2 * 1 * 1 = 24$
- Podemos usar essa definição para escrever uma função que computa o fatorial?

Fatorial

- Qual o valor de $n!$?
 - a multiplicação de todos os inteiros de 1 até n
 - $n * (n-1)!$. Além disso, $0! = 1$
- $4! = 4 * 3! = 4 * 3 * 2! = 4 * 3 * 2 * 1! = 4 * 3 * 2 * 1 * 0! = 4 * 3 * 2 * 1 * 1 = 24$
- ```
def fatorial(n):
 if n == 0:
 return 1
 return n * fatorial(n-1)
```

# O que acontece?

fatorial(4)

n = 4

# O que acontece?

$n = 4$

$n = 3$

# O que acontece?

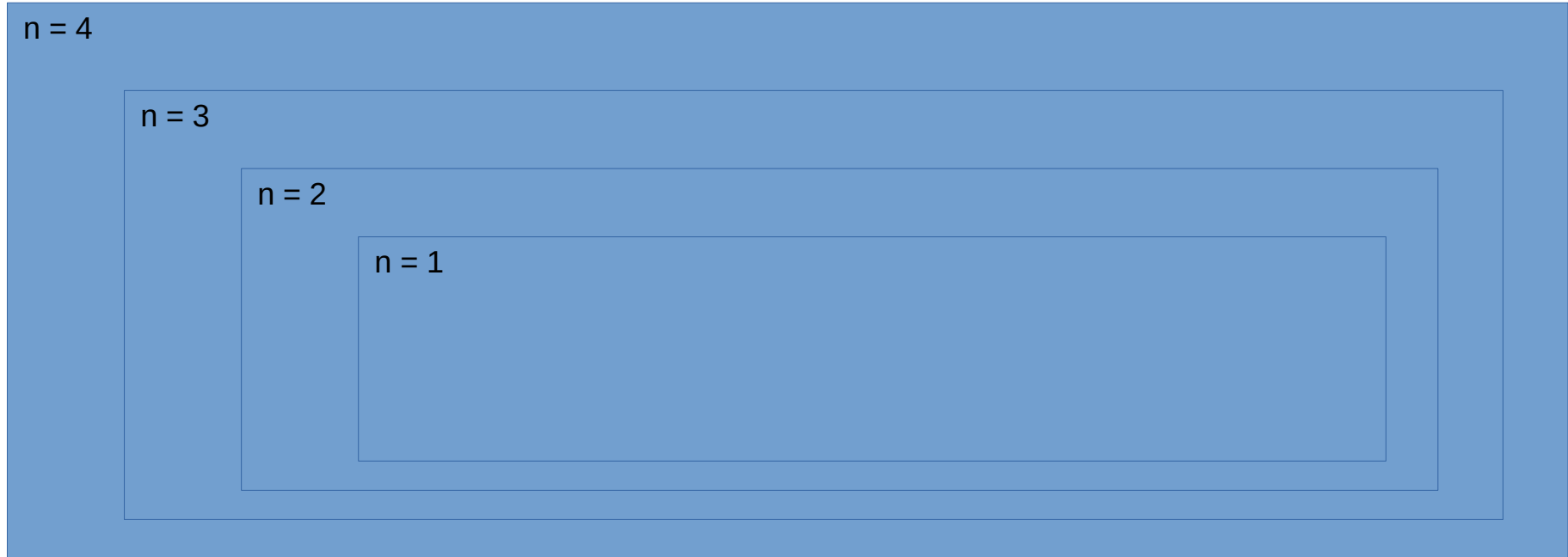
$n = 4$

$n = 3$

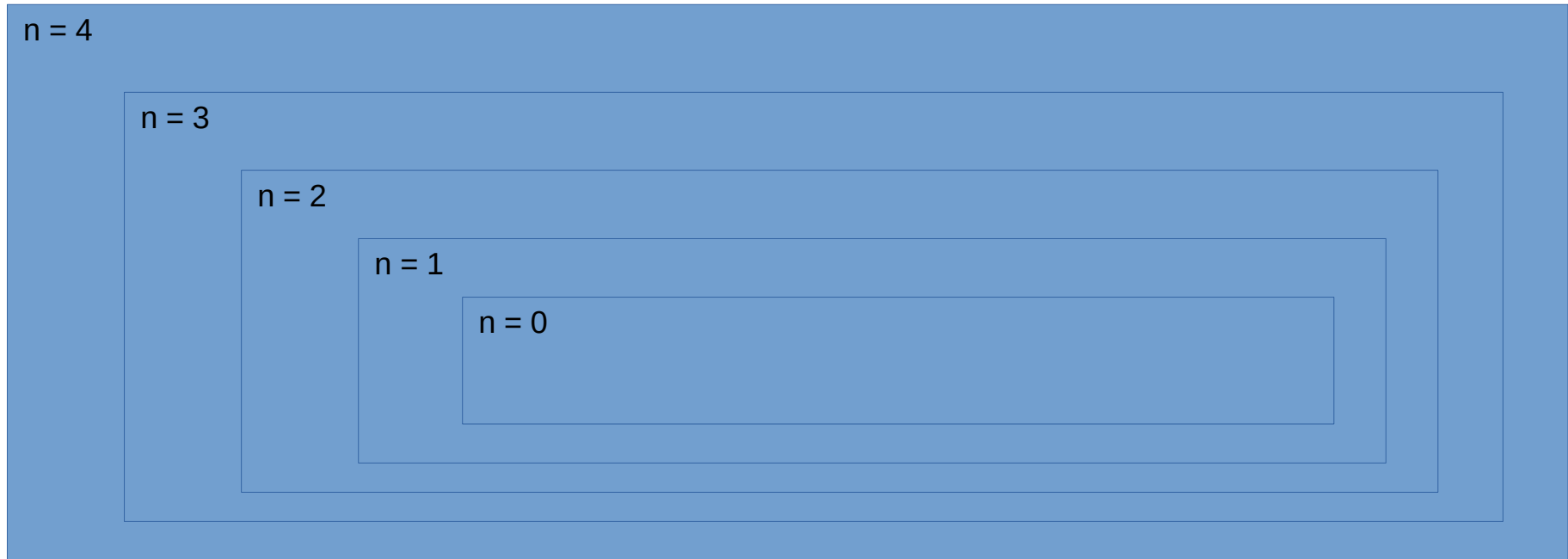
$n = 2$



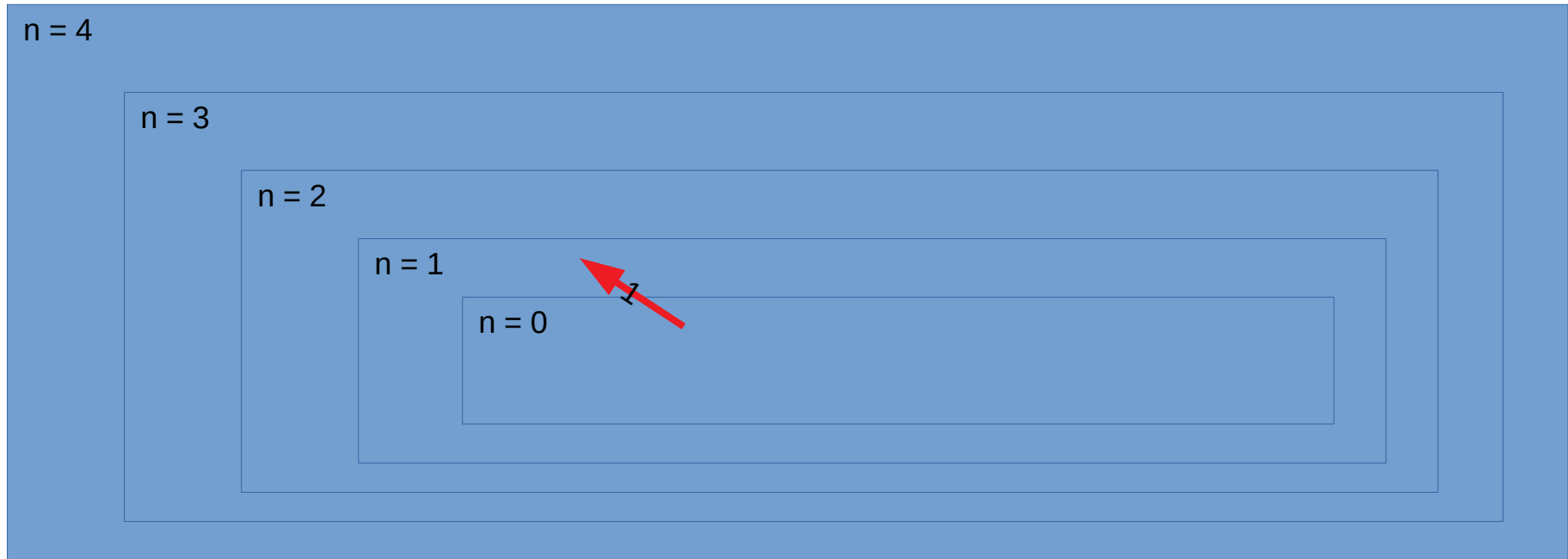
# O que acontece?



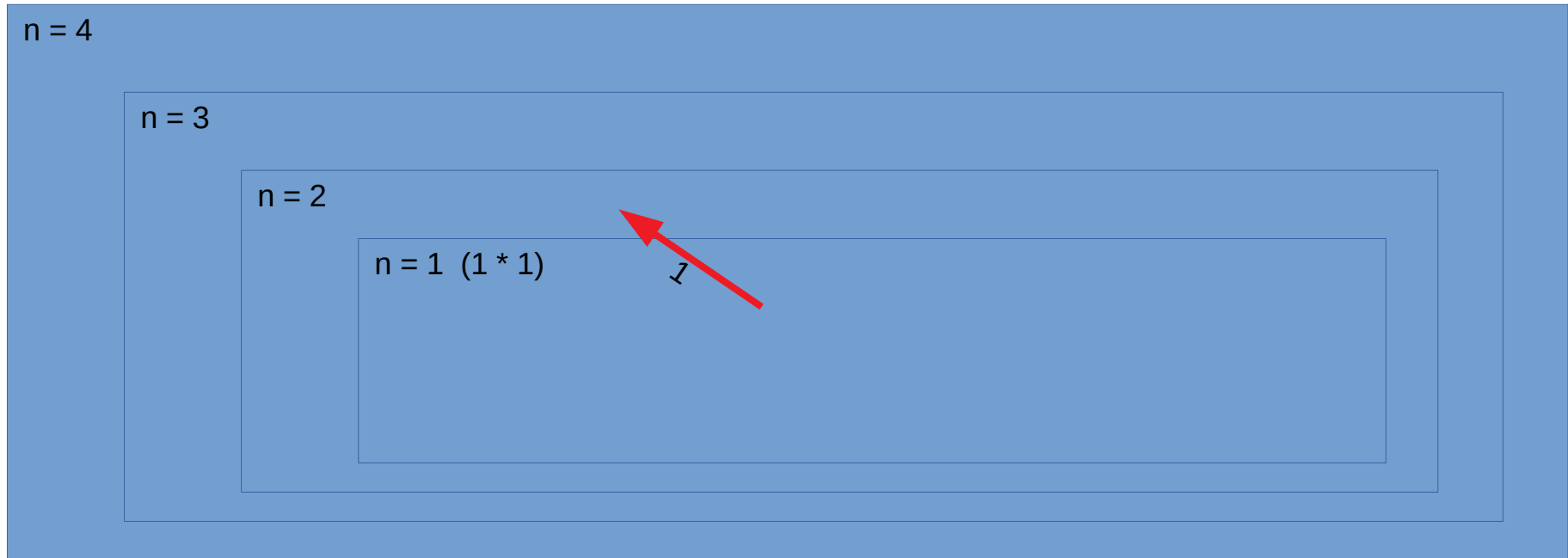
# O que acontece?



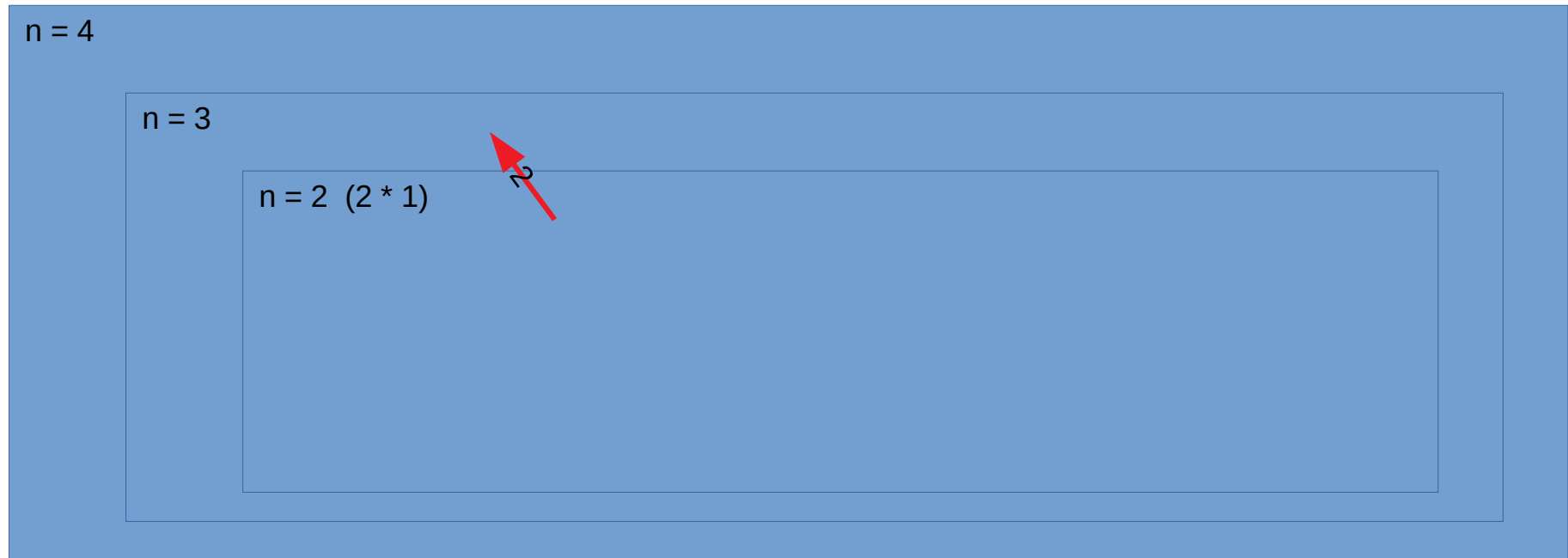
# O que acontece?



# O que acontece?



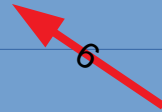
# O que acontece?



# O que acontece?

$n = 4$

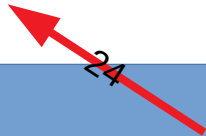
$n = 3 (3 * 2)$



6

# O que acontece?

$n = 4 \ (4 * 6)$



# Outro exemplo

- O máximo divisor comum (abreviadamente, MDC) entre dois ou mais números é o maior número que é fator de tais números. Por exemplo, os divisores comuns de 12 e 18 são 1,2,3 e 6, logo  $\text{mdc}(12, 18) = 6$ .



# Outro exemplo

- $\text{MDC}(x, x) = x$ ;
- $\text{MDC}(x, y) = \text{MDC}(x - y, y)$ , se  $x > y$ ;
- $\text{MDC}(x, y) = \text{MDC}(y, x)$ .
- Como implementar?

# Outro exemplo

- $\text{MDC}(x, x) = x;$   $\text{MDC}(76,76) = 76$
- $\text{MDC}(x, y) = \text{MDC}(x - y, y),$  se  $x > y;$   $\text{MDC}(76,20) = \text{MDC}(56,20)$
- $\text{MDC}(x, y) = \text{MDC}(y, x).$   $\text{MDC}(20,76) = \text{MDC}(76,20)$
- Como implementar?

# MDC

```
def mdc(x,y):
 if x == y:
 return x
 if x > y:
 return mdc(x-y, y)
 return mdc(y,x)
```

# Advertência

- Cada chamada de função, consome memória
- O interpretador vai limitar o número de chamadas

# Advertência

> python mdc.py

Entre com o 1o. valor: 1234567

Entre com o 2o. valor: 890

Traceback (most recent call last):

File "mdc.py", line 13, in <module>

print('O valor do MDC é: '.format(mdc(x,y)))

File "mdc.py", line 6, in mdc

return mdc(x-y, y)

[Previous line repeated 994 more times]

File "mdc.py", line 3, in mdc

if x == y:

RecursionError: **maximum recursion depth exceeded** in comparison

# Praticando

- 1. Crie uma função recursiva que receba um número inteiro  $N$  e calcule a soma dos números de 1 até  $N$ .
- 2. Escreva uma função recursiva para somar os elementos de uma lista de números. Ou seja, a função recebe uma lista como parâmetro e retorna um número, que é a soma dos elementos da lista.