

PROBLEMAS INVERSOS EM PYTHON

FUNÇÃO RECURSIVA

Profa.: Josiele da Silva Teixeira

Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - CEFET
Campus UNED Nova Friburgo

November 7, 2024

1. FUNÇÃO RECURSIVA

FUNÇÃO RECURSIVA

Definição: FUNÇÃO RECURSIVA

- * Uma função é dita **recursiva** quando dentro dela é feita uma ou mais chamadas a ela mesma.
- * A ideia é dividir um problema original em subproblemas menores de mesma natureza (divisão) e depois combinar as soluções obtidas para gerar a solução do problema original de tamanho maior (conquista).
- * Os subproblemas são resolvidos recursivamente do mesmo modo em função de instâncias menores, até se tornarem problemas triviais que são resolvidos de forma direta, interrompendo a recursão.
- * Todo esquema de recursividade exige uma condição de parada.
- * E o que acontece se não tivermos uma condição de parada? Ocorre um erro chamado de *stack overflow*, ou seja, estouro de pilha.

Exemplo: FUNÇÃO RECURSIVA

* PROBLEMA: Calcular o fatorial de um número.

=> **Solução 1:** Solução não recursiva.

```
1 | def fatorial(n):  
2 |     fat = 1  
3 |     while n > 1:  
4 |         fat *= n  
5 |         n -= 1  
6 |     return fat
```

Exemplo: FUNÇÃO RECURSIVA

* PROBLEMA: Calcular o fatorial de um número.

=> **Solução 2:** Solução recursiva: $n! = n.(n - 1)!$.

$$\text{fat}(n) = \begin{cases} 1 & \text{se } n=0 \\ n \times \text{fat}(n-1) & \text{se } n>0 \end{cases}$$

```
1 def fatorial(n):
2     if n == 0: #Caso trivial
3         return 1 #Solução direta
4     else:
5         return n*fatorial(n-1) #Chamada recursiva
```