

Recursão



Recursão

- Uma função é recursiva quando chama a si mesma.
- Exemplo:

```
def sum(n):  
    if (n<=1):  
        return n  
    return (n+sum(n-1))
```

Entrada	Saída	
Sum(1)	1	
Sum(2)	2 + sum(1)	2+ 1
Sum(3)	3 + sum(2)	3 + 2 + 1
Sum(4)	4 + sum(3)	4 + 3 + 2 + 1



Exemplo 1

Recursão

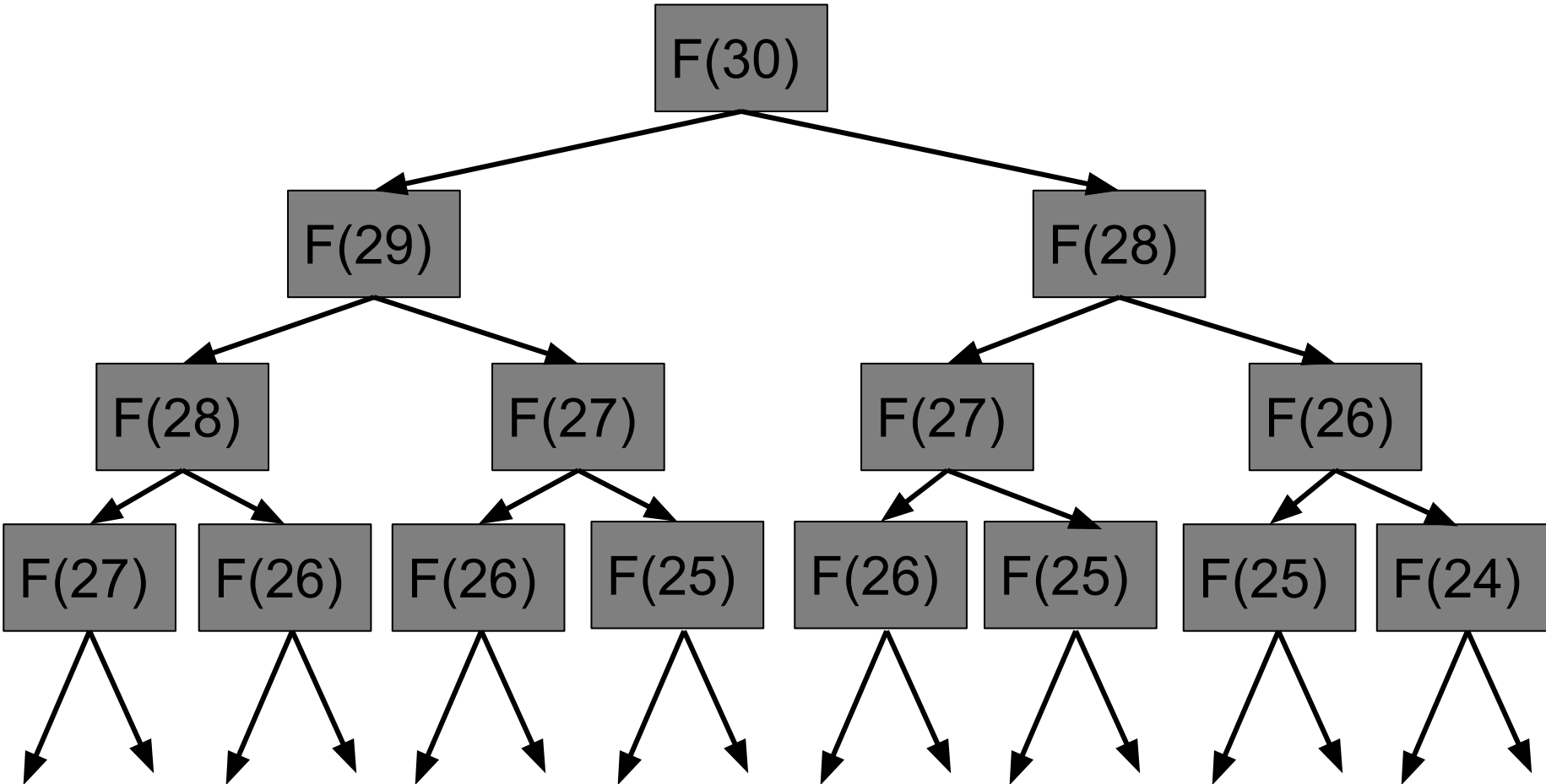
- Exemplo: Sequência de Fibonacci.

$$f(0)=0, f(1)=1, f(i+1)=f(i)+f(i-1), i=1,2,\dots$$

```
def fib(int n):  
    if(n<=1):  
        return n  
    else:  
        return (fib(n-1) + fib(n-2))
```

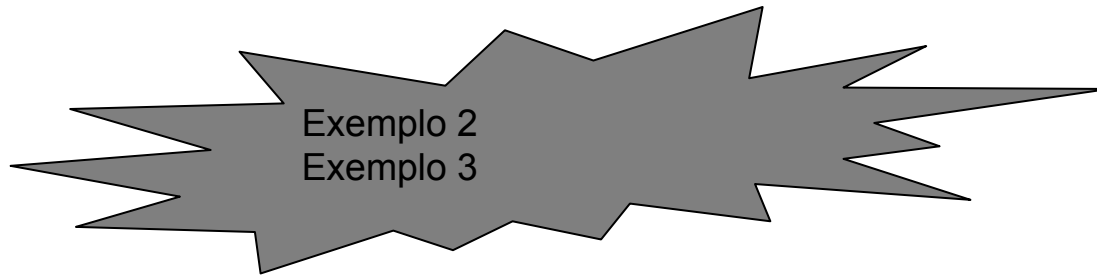
n	F(n)	Número de chamadas da função
0	1	1
1	1	1
2	1	3
...
23	28657	92735
24	46368	150049

Recursão



Recursão

- Recursão vs Iteração



Recursão

- Recursão vs Iteração
 - Iteração usa estrutura de repetição
 - Recursão usa estrutura de seleção (chamadas de funções repetitivas).
 - Iteração usa violação da condição como critério de parada.
 - Recursão usa passo base como critério de parada.
 - Ambos podem ser executados infinitamente, se o critério de parada não for satisfeito.
- Desvantagens da Recursão
 - Gera sobrecarga (overhead) com as chamadas de função, gerando gasto de tempo de processamento e espaço de memória.
 - Uma cópia da função (variáveis da função) é criada, consumindo memória.
 - Logo, a iteração tende a ser mais rápida por não fazer repetidas chamadas de funções.

Recursão

- Quando usar Recursão ou Iteração?

Recursão	Iteração
Estruturas condicionais	Estruturas de repetição
Repetição implícita	Repetição explícita
Caso base como critério de parada	Condição com critério de parada
Lento	Rápido
Solução simples	Solução complexa
Fácil manutenção	Difícil Manutenção

Recursão

Exemplo 4
Exemplo 5

Palíndromo

ANA

Arara rara

**Anotaram a data
da maratona**