

## Sobre Resolução de Recorrências com Teorema Mestre

Prof. M. Sc. Rodrigo Hagstrom

Exemplo com base em um programinha em Python:

```
def soma_lista(arr):  
    if len(arr) == 1:  
        return arr[0] # Caso base  
    meio = len(arr) // 2  
    esquerda = soma_lista(arr[:meio])  
    direita = soma_lista(arr[meio:])  
    return esquerda + direita
```

Análise pode ser feita com Teorema Mestre pois o tempo de execução deste algoritmo pode ser descrito pela recorrência:

$$T(n) = 2T(n/2) + O(1)$$

Onde:

- **a = 2** (duas chamadas recursivas)
- **b = 2** (o problema é dividido em duas partes de tamanho  $n/2$ )
- **f(n) = O(1)** (tempo de processamento fora das chamadas recursivas)

Agora, basta aplicar o Teorema Mestre para classificar a complexidade assintótica.