



# **Técnicas de Projeto de Algoritmos - Programação Dinâmica**

**Kleber Jacques F. de Souza**

# Programação Dinâmica

- Programação dinâmica é um método para a construção de algoritmos para a resolução de problemas computacionais, em especial os de **otimização combinatória**.
- Ela é aplicável a problemas nos quais a **solução ótima** pode ser **computada** a partir da solução ótima **previamente calculada e memorizada**.

# Programação Dinâmica

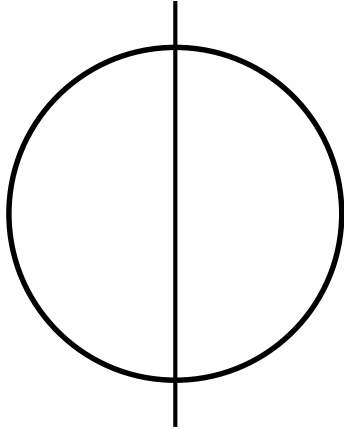
- Essa técnica de projeto de algoritmos pode ser definido, como “**recursão com apoio de uma tabela**”.
- Como em um algoritmo recursivo, cada instância do problema é resolvida a partir da solução de instâncias menores, ou melhor, de **sub-instâncias da instância original**.

# Programação Dinâmica

- A programação dinâmica, assim como a técnica de divisão e conquista, **resolve problemas combinando as soluções para subproblemas.**
  - Quando uma solução ótima para o problema contém em seu interior **soluções ótimas para subproblemas.**

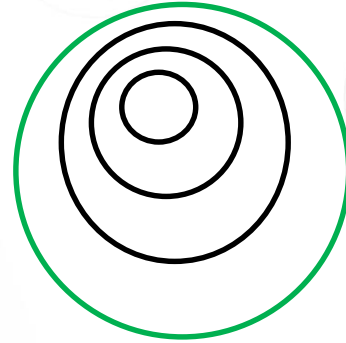
# Programação Dinâmica

Divisão e conquista



Abordagem top-down

Programação Dinâmica



Abordagem bottom-up

# Programação Dinâmica: Características

- 1. Subestrutura ótima:** Quando uma solução ótima para o problema contém em seu interior soluções ótimas para subproblemas.
- 2. Superposição de subproblemas:** Quando um algoritmo recursivo reexamina o mesmo problema muitas vezes.

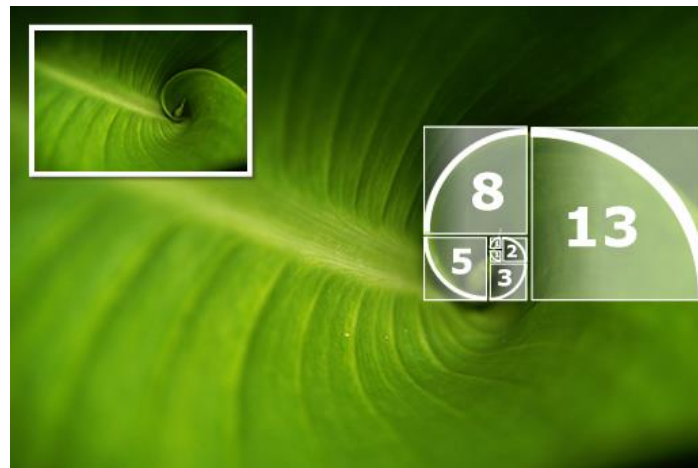
# Programação Dinâmica: Sequencia de Fibonacci

## Problema

Sequência (números) de Fibonacci:

$F = 1, 1, 2, 3, 5, 8, 13, 21, 34, \dots$

$$\text{Fib}(n) = \begin{cases} 1 & n \leq 2 \\ \text{Fib}(n-1) + \text{Fib}(n-2) & n > 2 \end{cases}$$



Wikimedia, 2018

PUC Minas Virtual

# Programação Dinâmica: Sequencia de Fibonacci

```
int Fib(int n)
{
    if (n <= 2) return 1;

    return Fib(n - 1) + Fib(n - 2);
}
```

- Quando a forma **recursiva** é implementada sem maiores cuidados, **sem memorização**, o seu cálculo de tempo **cresce exponencialmente**.



# Programação Dinâmica: Sequencia de Fibonacci

```
int fib_PD(int n)
{
    int[] memory = new int[n + 1];

    memory[1] = 1;
    memory[2] = 1;

    for (int i = 3; i <= n; i++) {
        memory[i] = memory[i - 1] + memory[i - 2];
    }

    return memory[n];
}
```

# Referências Bibliográficas

Ziviani, Nivio. **Projeto de Algoritmos:** com implementações em JAVA e C++. CENGAGE Learning, 2012. (Livro Eletrônico)

CORMEN, Thomas H. et al. **Algoritmos:** teoria e prática. Elsevier, RJ, 2012.

Wikimedia, 2018. Disponível em:  
<https://commons.wikimedia.org/wiki/File:Bromelia.png>.