

Iteração e Recурсão

- Definições
- Exemplos
- Programação
(Laços, condicional, etc.)

Definições

Iteração ocorre quando o mesmo procedimento é repetido várias vezes.

Recursão ocorre quando o resultado (saída) de uma iteração se torna o argumento (entrada) da próxima.

Exemplo de iteração

Como determinar se um número n é primo?

0. Comece com o número 2.
1. Divida
2. Verifique se o resto da divisão é zero.
 - a. se sim, pare. Então n não é primo.
 - b. se não, vá para o passo 3.
3. Se foi o último número, pare. Então n é primo.
4. Tome o próximo número e vá para o passo 1.

Exercício 1. Implemente essa rotina em uma linguagem

Exemplo de iteração com recursão

0. Considere o número 325.
1. Se o número é par, divida ele por 2.
Se o número é ímpar, multiplique por 3 e adicione 1.
2. Repita até que o número se torne 1.

Conte o número de iterações ("dividir por 2" e "multiplicar por 3 e adicionar 1" são iterações

Exercício 2. Implemente a rotina acima. Imprima o número de iterações para 325 e 103, por exemplo.

Exemplo: Sequência de Fibonacci

$$x_0 = 0, \quad x_1 = 1$$

$$x_n = x_{n-1} + x_{n-2}$$

Sage Cell Server sagecell.sagemath.org

SageMathCell

Type some Sage code below and press Evaluate.

```
1+ def fibonacci_iterativo(n):
2+     a, b = 0, 1
3+     #print(str(a) + ", ", end="")
4+     for i in range(n):
5+         a, b = b, a + b
6+         #print(str(a) + ", ", end="")
7+     return a
8+
9+ fibonacci_iterativo(15)
```

Evaluate

610

Sage Cell Server sagecell.sagemath.org

SageMathCell

Type some Sage code below and press Evaluate.

```
1+ def fibonacci_recuritivo(n):
2+     if n == 0:
3+         return 0
4+     elif n == 1:
5+         return 1
6+     else:
7+         return fibonacci_recuritivo(n-1) + fibonacci_re
8+
9+ fibonacci_recuritivo(15)
```

Evaluate

610

Navigation icons: back, forward, search, etc.