Algoritmos e Estruturas de Dados

Recursividade

Autores:
Carlos Urbano
Catarina Reis
José Magno
Marco Ferreira

Indução Matemática

- O **Método de Indução Matemática** é uma metodologia de demonstração baseada no Princípio de Indução Finita, através dos passos seguintes:
  - **Base de indução**: Estabelecer a propriedade para o primeiro número natural, ou seja, o número 1
  - **Passo de indução**: Estabelecer que caso a propriedade se verifique para um número natural n (Hipótese de Indução) então ela também é verificada para o número natural seguinte, $n + 1$

Indução Matemática

- Assim para demonstrarmos que uma propriedade
  - $P(n)$ é verdadeira $\forall_n \in \mathbb{N}$,

devemos provar que:

  - $P(1)$ é verdadeira
  - $P(n) \Rightarrow P(n+1)$

Indução Matemática - Exemplo

- Números Quadrados de Pitágoras
  - Sendo $\mathrm{i}_n = 1 + \cdots + (2n - 1) = \sum_{k=1}^{n}(2k - 1), \forall_n \in \mathbb{N}$
  - Pretende-se demonstrar que $\forall_n \in \mathbb{N}, i_n = n^2$

Indução Matemática - Exemplo

- Números Quadrados de Pitágoras
  - Sendo $\mathrm{i}_n = 1 + \cdots + (2n-1) = \sum_{k=1}^{n}(2k-1), \forall_n \in \mathbb{N}$
  - Pretende-se demonstrar que $\forall_n \in \mathbb{N}, i_n = n^2$

Por observação geométrica

n=1
1
2x1-1
1^2

Indução Matemática - Exemplo

- Números Quadrados de Pitágoras
  - Sendo $i_n = 1 + \cdots + (2n-1) = \sum_{k=1}^{n}(2k-1), \forall_n \in \mathbb{N}$
  - Pretende-se demonstrar que $\forall_n \in \mathbb{N}, i_n = n^2$

Por observação geométrica

n=1	n=2
1	1+3=4
2x1-1	1+(2x2-1)
1^2	2^2

Indução Matemática - Exemplo

- Números Quadrados de Pitágoras
  - Sendo $i_n = 1 + \cdots + (2n-1) = \sum_{k=1}^{n}(2k-1), \forall_n \in \mathbb{N}$
  - Pretende-se demonstrar que $\forall_n \in \mathbb{N}, i_n = n^2$

Por observação geométrica

n=1	n=2	n=3
1	1+3=4	4+5=9
2x1-1	1+(2x2-1)	4+(2x3-1)
1^2	2^2	3^2

Indução Matemática - Exemplo

- Números Quadrados de Pitágoras
  - Sendo $\mathrm{i}_n = 1 + \cdots + (2n - 1) = \sum_{k=1}^{n}(2k - 1), \forall_n \in \mathbb{N}$
  - Pretende-se demonstrar que $\forall_n \in \mathbb{N}, i_n = n^2$

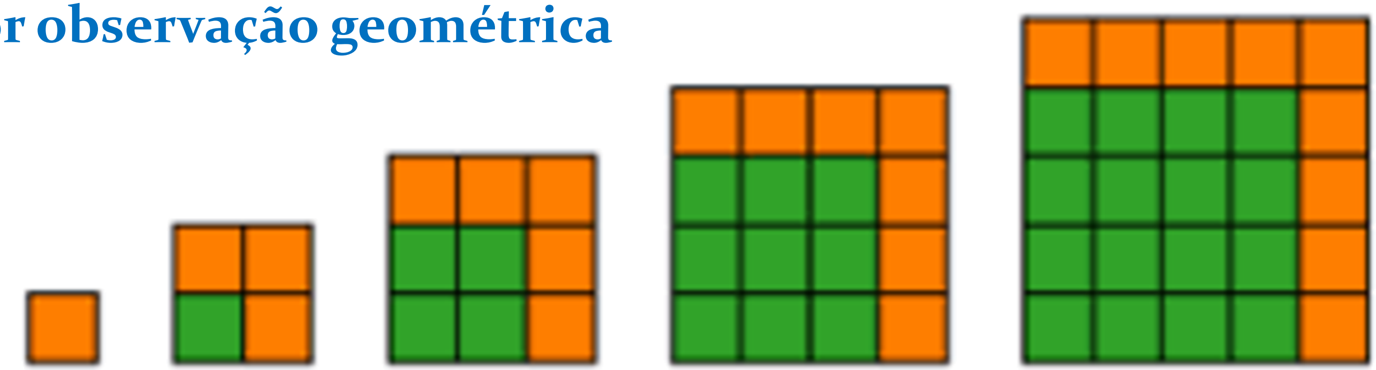
Por observação geométrica

n=1	n=2	n=3	n=4
1	1+3=4	4+5=9	9+7=16
2x1-1	1+(2x2-1)	4+(2x3-1)	9+(2x4-1)
1^2	2^2	3^2	4^2

Indução Matemática - Exemplo

- Números Quadrados de Pitágoras
  - Sendo $i_n = 1 + \cdots + (2n-1) = \sum_{k=1}^{n}(2k-1), \forall_n \in \mathbb{N}$
  - Pretende-se demonstrar que $\forall_n \in \mathbb{N}, i_n = n^2$

Por observação geométrica

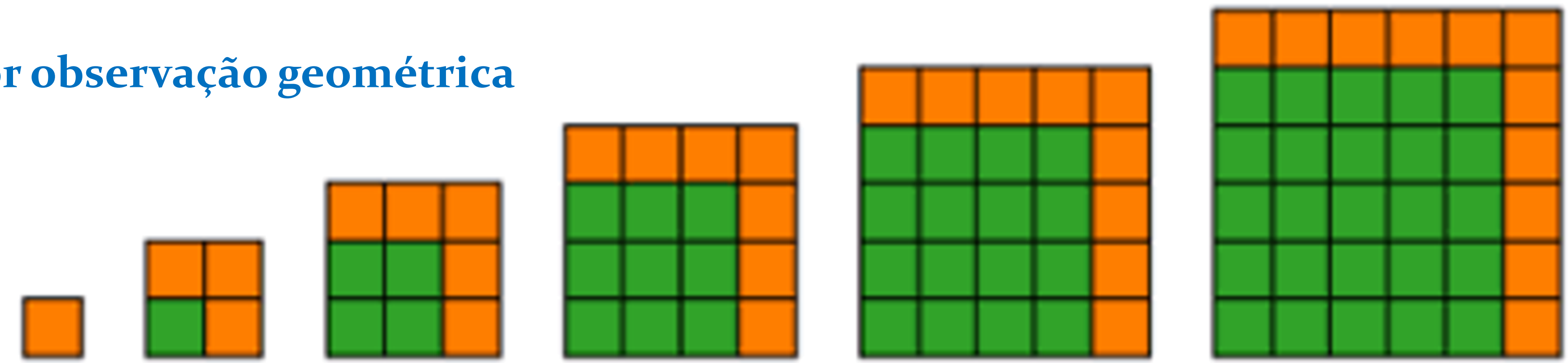


n=1	n=2	n=3	n=4	n=5
1	1+3=4	4+5=9	9+7=16	16+9=25
2x1-1	1+(2x2-1)	4+(2x3-1)	9+(2x4-1)	16+(2x5-1)
1^2	2^2	3^2	4^2	5^2

Indução Matemática - Exemplo

- Números Quadrados de Pitágoras
  - Sendo $i_n = 1 + \cdots + (2n-1) = \sum_{k=1}^{n}(2k-1), \forall_n \in \mathbb{N}$
  - Pretende-se demonstrar que $\forall_n \in \mathbb{N}, i_n = n^2$

Por observação geométrica



n=1	n=2	n=3	n=4	n=5	n=6
1	1+3=4	4+5=9	9+7=16	16+9=25	25+11=36
2x1-1	1+(2x2-1)	4+(2x3-1)	9+(2x4-1)	16+(2x5-1)	25+(2x6-1)
1^2	2^2	3^2	4^2	5^2	6^2

Indução Matemática - Exemplo

- Números Quadrados de Pitágoras
  - Sendo $\mathrm{i}_n = 1 + \cdots + (2n - 1) = \sum_{k=1}^{n}(2k - 1), \forall_n \in \mathbb{N}$
  - Pretende-se demonstrar que $\forall_n \in \mathbb{N}, i_n = n^2$
- Assim, por Indução Matemática temos que:
  - $P(1)$ é verdadeira, pois $1 = 1^2$

Indução Matemática - Exemplo

- Números Quadrados de Pitágoras
  - Sendo $i_n = 1 + \cdots + (2n - 1) = \sum_{k=1}^{n}(2k - 1), \forall_n \in \mathbb{N}$
  - Pretende-se demonstrar que $\forall_n \in \mathbb{N}, i_n = n^2$
- Assim, por Indução Matemática temos que:
  - $P(1)$ é verdadeira, pois $1 = 1^2$
  - Supondo que $P(n) : \ i_n = n^2$ é verdadeira, vamos provar que $P(n+1) : i_{n+1} = (n+1)^2$

Indução Matemática - Exemplo

- Supondo que $P(n)$: $i_n = n^2$ é verdadeira, vamos provar que $P(n+1) : i_{n+1} = (n+1)^2$

Indução Matemática - Exemplo

- Supondo que $P(n)$: $i_n = n^2$ é verdadeira, vamos provar que

$P(n+1) : i_{n+1} = (n+1)^2$

$$i_{n+1} = 1 + \cdots + (2n-1) + (2(n+1)-1)$$

Indução Matemática - Exemplo

- Supondo que $P(n): \ i_n = n^2$ é verdadeira, vamos provar que $P(n+1): i_{n+1} = (n+1)^2$

$$
\begin{aligned}
i_{n+1} &= 1 + \cdots + (2n-1) + (2(n+1)-1) \\
&= i_n + (2(n+1)-1)
\end{aligned}
$$

Indução Matemática - Exemplo

- Supondo que $P(n): \; i_n = n^2$ é verdadeira, vamos provar que

$P(n+1): i_{n+1} = (n+1)^2$

$$i_{n+1} = 1 + \cdots + (2n-1) + (2(n+1)-1)$$

$$= i_n + (2(n+1)-1)$$

$$= i_n + (2n+2-1)$$

Indução Matemática - Exemplo

- Supondo que $P(n)$: $i_n = n^2$ é verdadeira, vamos provar que $P(n+1) : i_{n+1} = (n+1)^2$

$$
\begin{aligned}
i_{n+1} &= 1 + \cdots + (2n-1) + (2(n+1)-1) \\
&= i_n + (2(n+1)-1) \\
&= i_n + (2n+2-1) \\
&= i_n + 2n + 1
\end{aligned}
$$

Indução Matemática - Exemplo

- Supondo que $P(n):\ i_n = n^2$ é verdadeira, vamos provar que $P(n+1): i_{n+1} = (n+1)^2$

$$i_{n+1} = 1 + \cdots + (2n-1) + (2(n+1)-1)$$

$$= i_n + (2(n+1)-1)$$

$$= i_n + (2n + 2 - 1)$$

$$= i_n + 2n + 1$$

Usando a hipótese de indução $P(n)$

$$= n^2 + 2n + 1$$

Indução Matemática - Exemplo

- Supondo que $P(n)$: $i_n = n^2$ é verdadeira, vamos provar que

$P(n+1) : i_{n+1} = (n+1)^2$

$$
\begin{aligned}
i_{n+1} &= 1 + \cdots + (2n-1) + (2(n+1)-1) \\
&= i_n + (2(n+1)-1) \\
&= i_n + (2n+2-1) \\
&= i_n + 2n + 1
\end{aligned}
$$

Usando a hipótese de indução $P(n)$

$$
\begin{aligned}
&= n^2 + 2n + 1 \\
&= (n+1)^2
\end{aligned}
$$

Indução Matemática - Exemplo

- Supondo que $P(n)$: $i_n = n^2$ é verdadeira, vamos provar que $P(n+1)$: $i_{n+1} = (n+1)^2$

$$
\begin{aligned}
i_{n+1} &= 1 + \cdots + (2n-1) + (2(n+1)-1) \\
&= i_n + (2(n+1)-1) \\
&= i_n + (2n+2-1) \\
&= i_n + 2n + 1
\end{aligned}
$$

Usando a hipótese de indução $P(n)$

$$
\begin{aligned}
&= n^2 + 2n + 1 \\
&= (n+1)^2
\end{aligned}
$$

Logo $P(n) \Rightarrow P(n+1)$

Indução Matemática - Exercícios

Provar por indução matemática que:

- $9^n - 1$ é múltiplo de 8, $\forall_n \in \mathbb{N}$
- $2^{3n} - 1$ é múltiplo de 7, $\forall_n \in \mathbb{N}$

Indução Matemática - Exercícios

Recursividade

- Como se sabe, uma das maneiras de alcançar **repetições** é através de ciclos como **`for`** ou **`while`**

- Outro modo, é recorrendo à **recursividade**
  - Ocorre quando uma função se chama a si própria, direta ou indiretamente

Recursividade

- Exemplo: cálculo do fatorial
  - Definição:
    - O fatorial de um inteiro positivo n denota-se por n!, sendo definido pelo produto dos inteiros de 1 a n
    - Se n = 0 então , por convenção, 0! é definido pelo valor 1
  - Mais formalmente:

$$n! = \begin{cases} 1 & \Leftarrow \quad n = 0 \\ n(n-1)(n-2)\ \ldots\ 3 * 2 * 1 & \Leftarrow \quad n \in \mathbb{N} \end{cases}$$

Recursividade

- Exemplo: cálculo do fatorial
  - A função fatorial pode ser definida de maneira a sugerir uma formulação recursiva. Vejamos,
    - `fatorial(5) = 5*(4*3*2*1) = 5 * fatorial(4)`
  - Pode, assim, definir-se o `fatorial(5)` em termos do `fatorial(4)`
  - De uma forma geral, para um inteiro positivo n
    - `fatorial(n) = n * fatorial(n-1)`
  - Originando a seguinte definição **recursiva**

$$fatorial(n) = \begin{cases} 1 & \Leftarrow \quad n = 0 \\ n * fatorial(n-1) & \Leftarrow \quad n \in \mathbb{N} \end{cases}$$

Recursividade

- Um dos aspetos importantes numa definição recursiva é a indicação dos chamados **casos base**. São definidos de forma não-recursiva em termos de um valor fixo
- Existem também os **casos recursivos**, recorrendo à definição da função a ser definida de modo a convergir para um dos **casos base** evitando, assim, a recursividade infinita

- No exemplo da função fatorial,
  - o caso base é $n = 0$
  - no caso recursivo, cada vez que a função fatorial é invocada, o seu argumento n é decrementado de uma unidade, convergindo, assim, para o caso base $n = 0$

Recursividade

- Implementação da função fatorial

```
public int fatorial(int n) {
  if (n == 0) {
    return 1; //caso base
  }
  return n * fatorial(n - 1); //caso recursivo
}
```

Recursividade

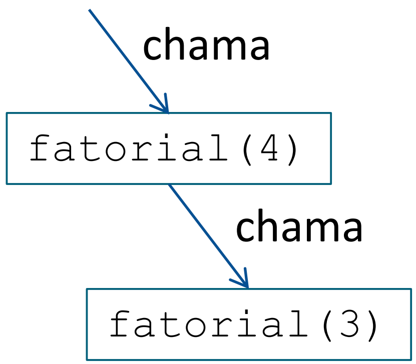
- Execução passo-a-passo de **`fatorial(4)`**

chama

```
fatorial(4)
```

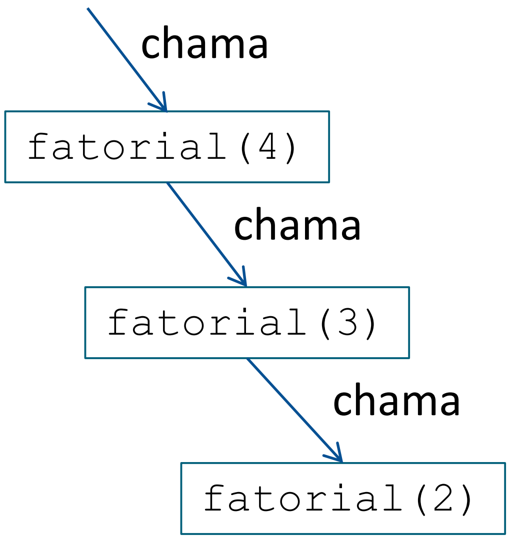
Recursividade

- Execução passo-a-passo de **`fatorial(4)`**



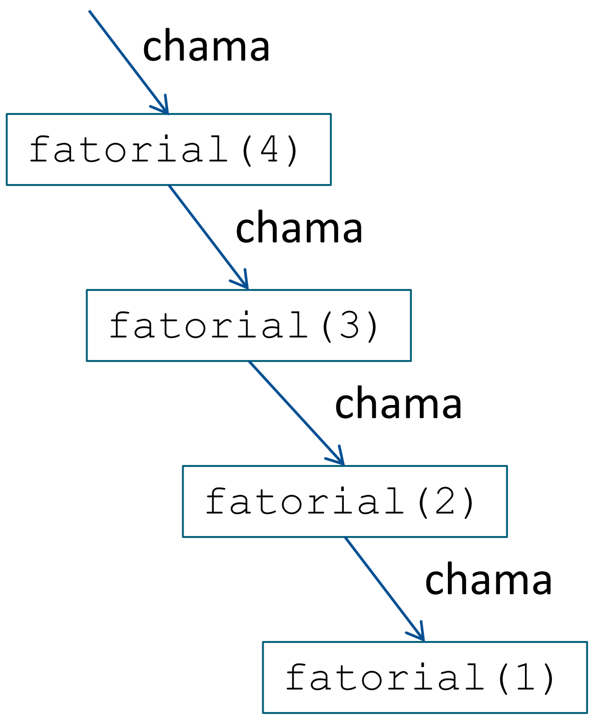
Recursividade

- Execução passo-a-passo de `fatorial(4)`



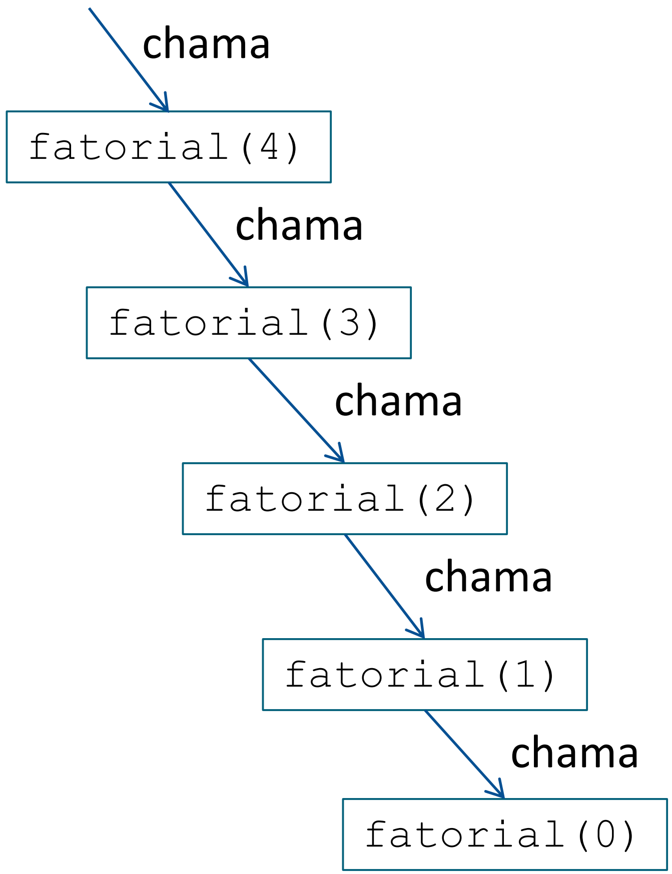
Recursividade

- Execução passo-a-passo de **`fatorial(4)`**



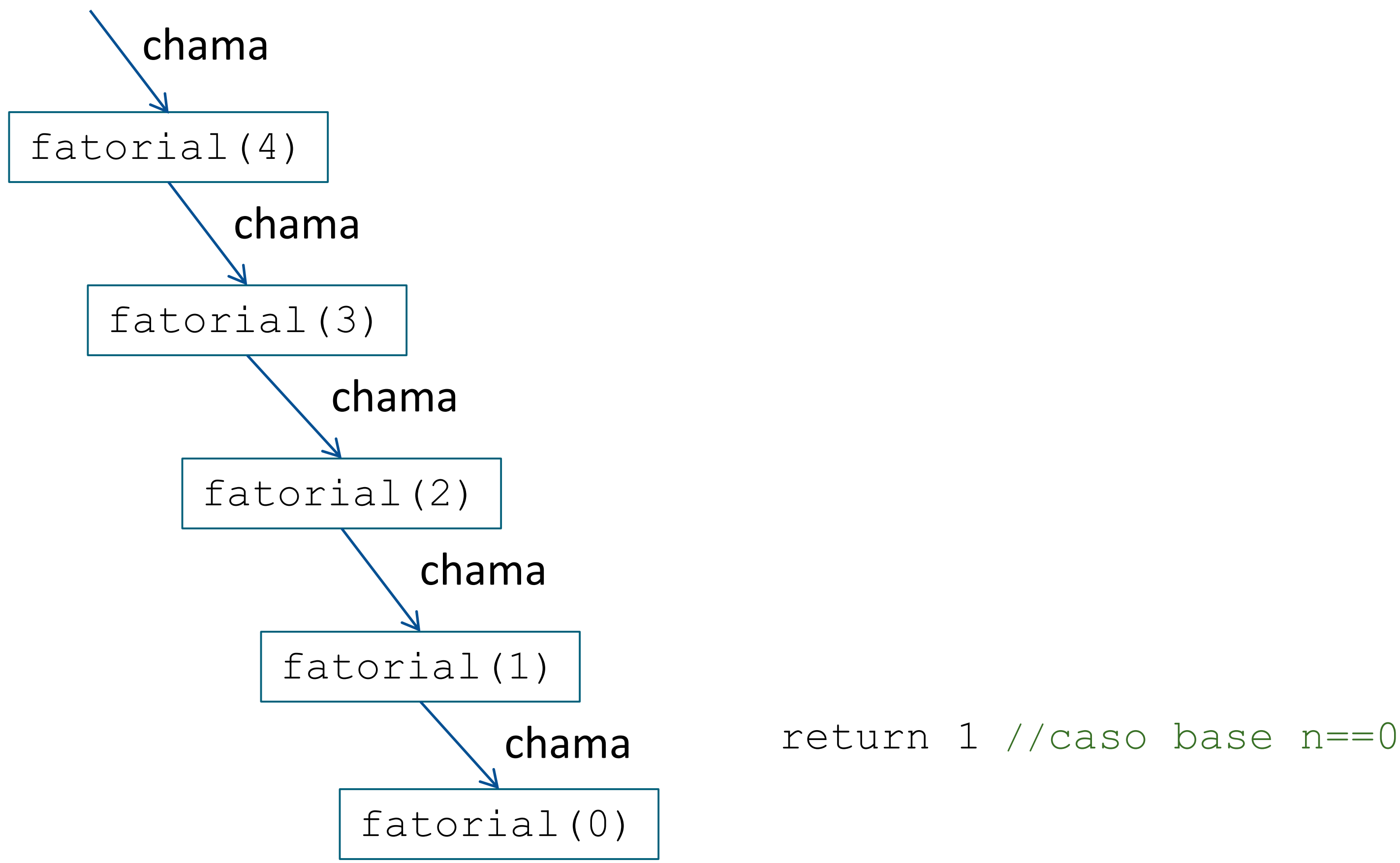
Recursividade

- Execução passo-a-passo de **`fatorial(4)`**



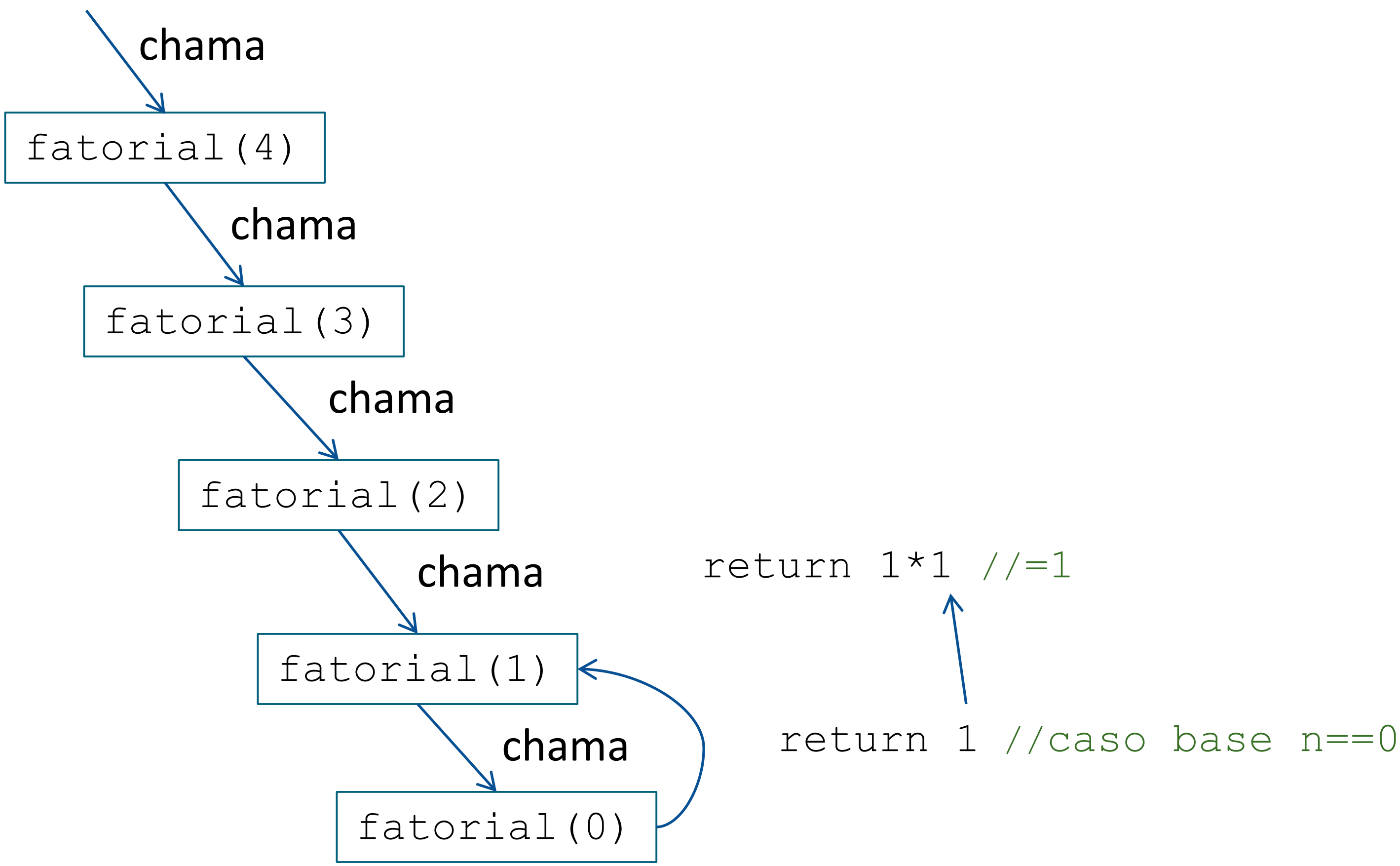
Recursividade

- Execução passo-a-passo de **`fatorial(4)`**



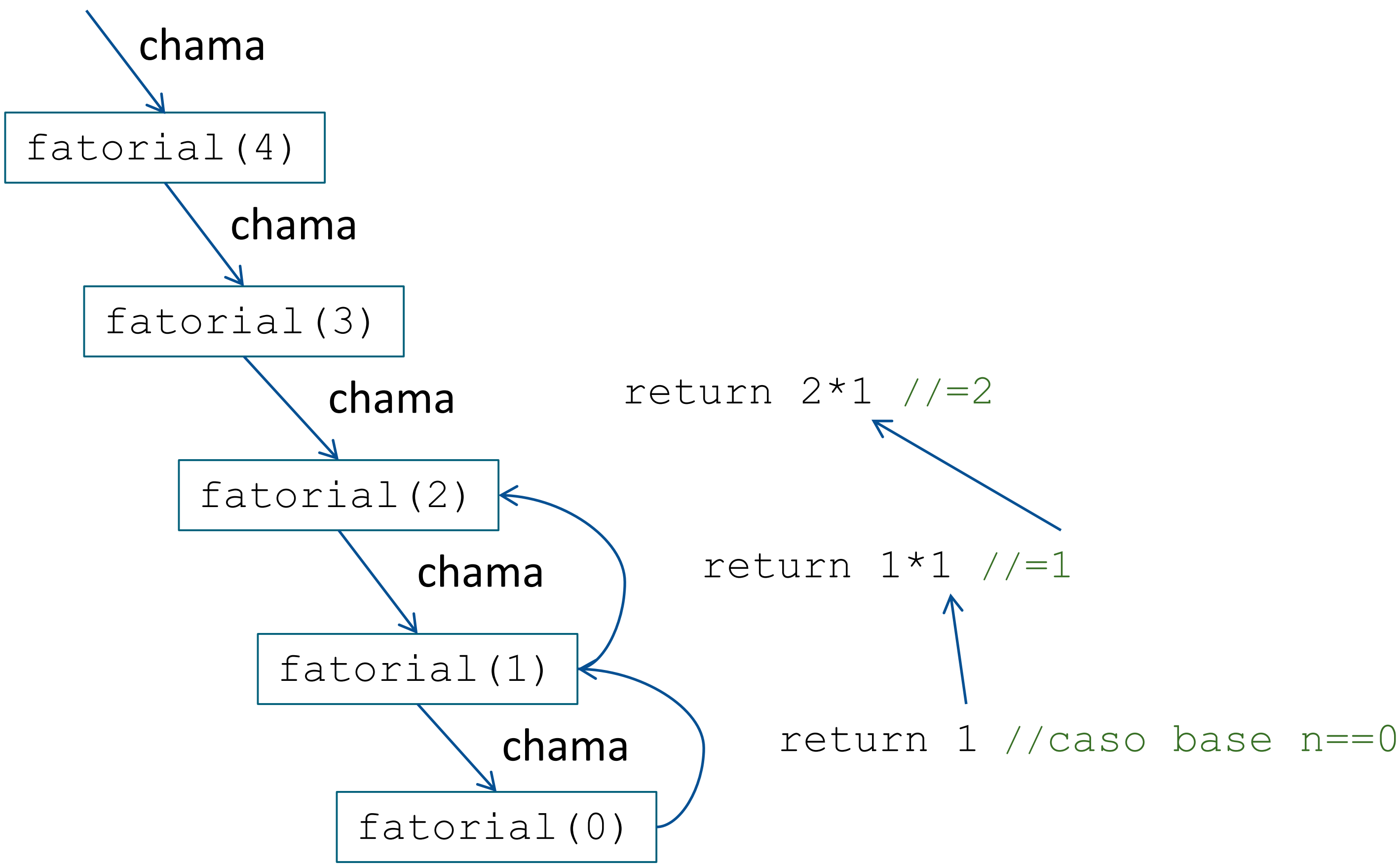
Recursividade

- Execução passo-a-passo de **`fatorial(4)`**



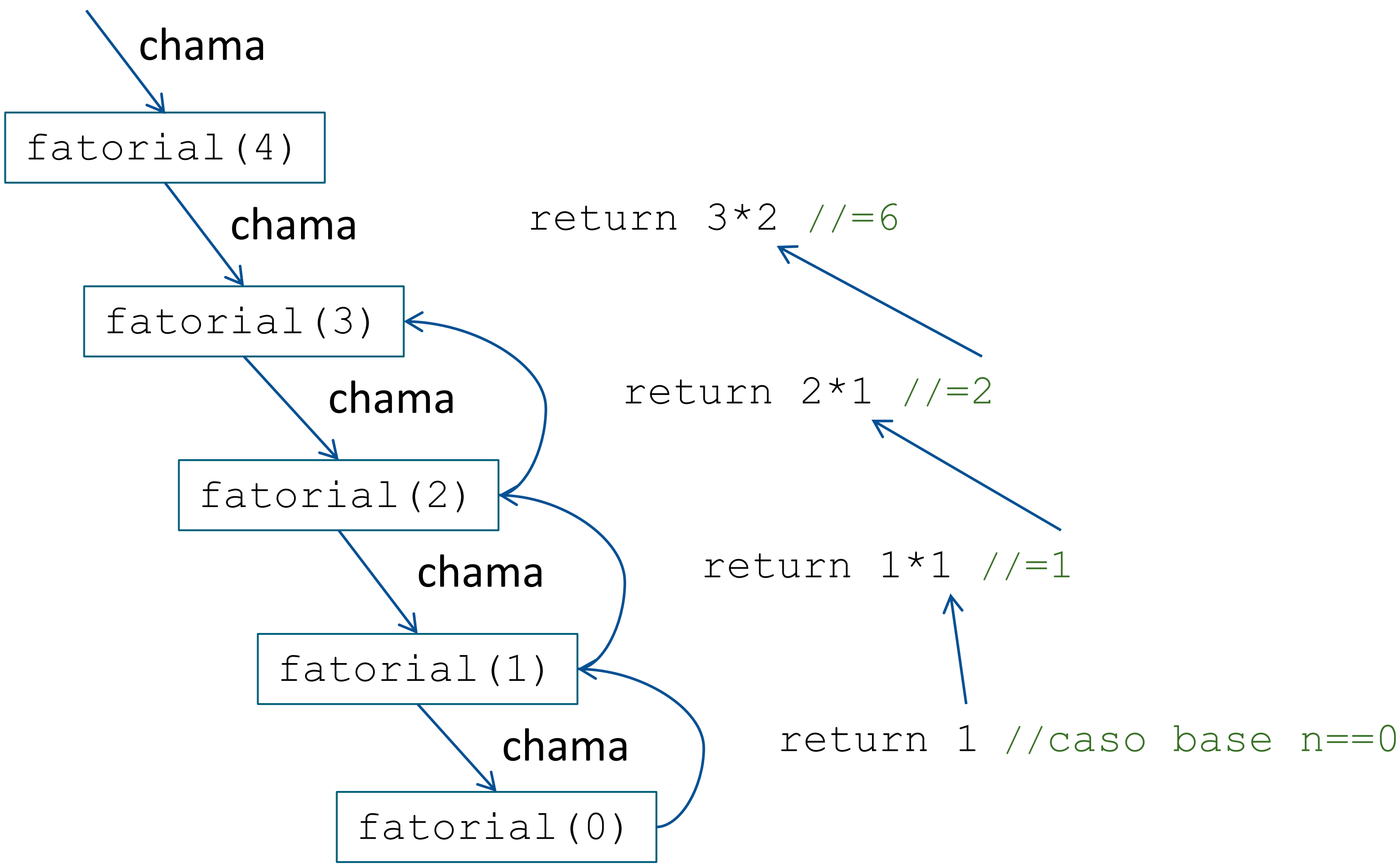
Recursividade

- Execução passo-a-passo de **`fatorial(4)`**



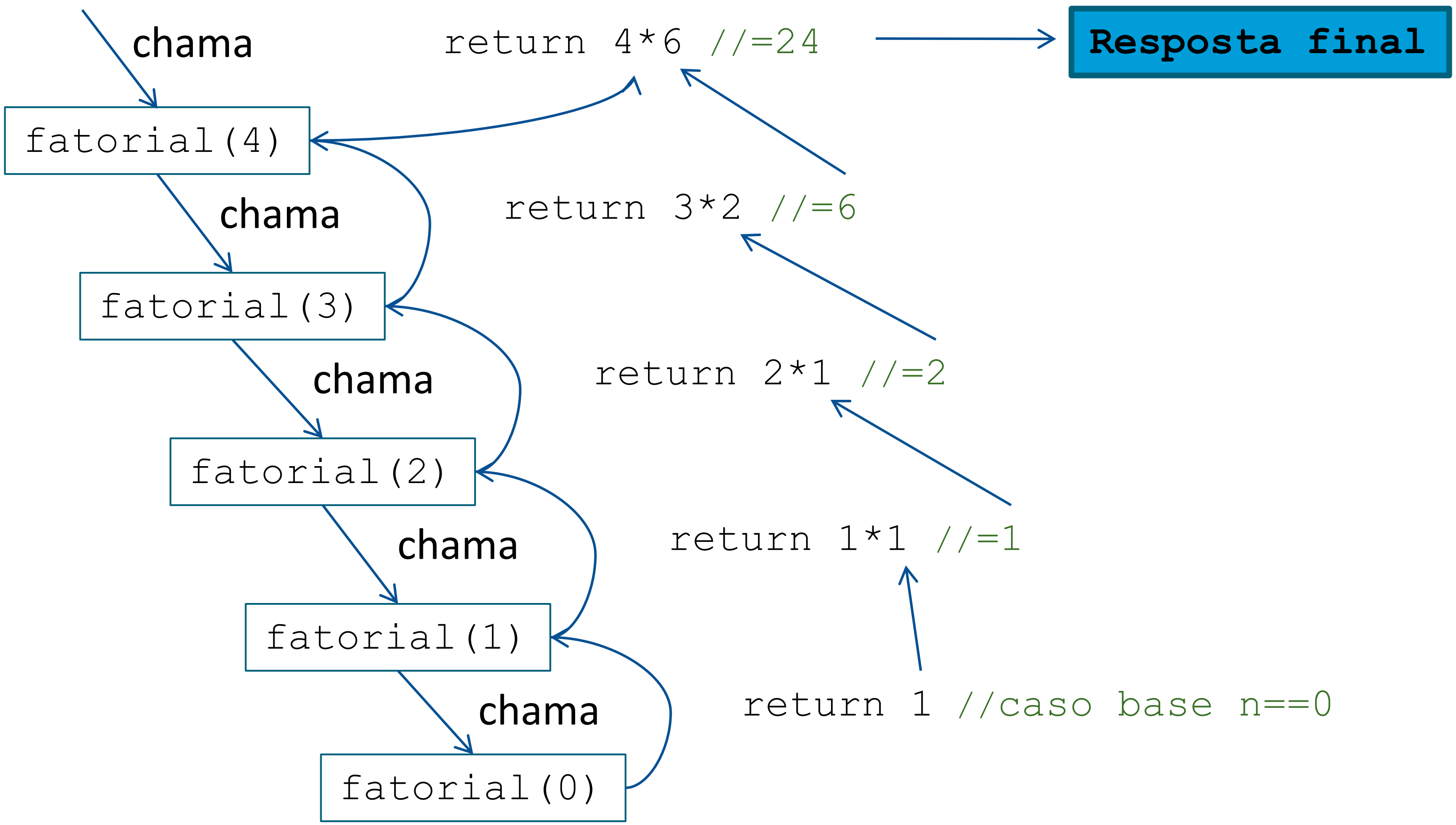
Recursividade

- Execução passo-a-passo de **`fatorial(4)`**



Recursividade

- Execução passo-a-passo de **`fatorial(4)`**



Recursividade

- Tipos de recursividade direta
  - Recursividade linear
    - Apenas uma chamada recursiva ao método
      - Ex: função fatorial
  - Recursividade binária
    - Duas chamadas recursivas ao método
      - Ex: funções nas árvores binárias
      - [sempre que um problema se pode dividir em 2 “metades”]
  - Recursividade múltipla
    - Diversas chamadas recursivas ao método
      - Ex: funções para preenchimento de polígonos (cima, baixo, esquerda, direita)

Recursividade

- Conclusões
  - Vantagens
    - Código de fácil leitura e compreensão
    - Código mais compacto
  - Desvantagens
    - A chamada de métodos é uma operação “cara” que envolve uma certa sobrecarga
      - Implica transferir o controlo para outro local
      - Implica guardar os argumentos do método e o endereço para retornar, colocando-os numa pilha interna
    - É necessário ocupar memória para guardar os argumentos intermédios e os valores de retorno

=> Pouco eficiente para problemas envolvendo grandes quantidades de dados

Recursividade - Exercícios

Implemente funções recursivas para:

- `somar(int n)` que efetue a soma dos inteiros de 1 a n, $\forall n \in \mathbb{N}$
- `potenciaDe2(int n)` que calcule 2^n, $\forall n \in \mathbb{N}_0$

Recursividade - Exercícios