

Funções Recursivas

Instituto de Computação - UNICAMP

Funções Recursivas: Exemplos

- Fatorial
- Soma
- Máximo divisor comum
- Sequência de Fibonacci
- Exponenciação
- Máximo de um vetor

$$n! = 1 \times 2 \times 3 \cdots \times n, \quad 0! = 1$$

$$n! = 1 \times 2 \times 3 \cdots \times n, \quad 0! = 1$$

fatorial($n - 1$) — — — — — $>$ *fatorial*(n)?

$$n! = 1 \times 2 \times 3 \cdots \times n, \quad 0! = 1$$

fatorial($n - 1$) — — — — — $>$ *fatorial*(n)?

$$\textit{fatorial}(n) = n * \textit{fatorial}(n - 1);$$

$$\textit{fatorial}(0) = 1.$$

Fatorial

$$n! = 1 \times 2 \times 3 \cdots \times n, \quad 0! = 1$$

$$\textit{fatorial}(n-1) \text{ --- } > \textit{fatorial}(n)?$$

$$\textit{fatorial}(n) = n * \textit{fatorial}(n-1);$$

$$\textit{fatorial}(0) = 1.$$

$$\begin{aligned}\textit{fatorial}(6) &= 6 * \textit{fatorial}(5) \\ &= 6 * (5 * \textit{fatorial}(4)) \\ &= 6 * (5 * (4 * \textit{fatorial}(3))) \\ &= 6 * (5 * (4 * (3 * \textit{fatorial}(2)))) \\ &= 6 * (5 * (4 * (3 * (2 * \textit{fatorial}(1))))) \\ &= 6 * (5 * (4 * (3 * (2 * (1 * \textit{fatorial}(0)))))) \\ &= 6 * (5 * (4 * (3 * (2 * (1 * 1))))) \\ &= 720.\end{aligned}$$

$$\begin{aligned}fatorial(0) &= 1; \\fatorial(n) &= n * fatorial(n - 1).\end{aligned}$$

```
int fatorial(int n){  
    if (n == 0) return 1;  
    return n * fatorial(n-1);  
}
```

Problema: calcular a soma $1 + 2 + 3 + \cdots + n - 1 + n$

Problema: calcular a soma $1 + 2 + 3 + \cdots + n - 1 + n$

$$soma(n) = n + soma(n - 1);$$

$$soma(1) = 1.$$

Problema: calcular a soma $1 + 2 + 3 + \dots + n - 1 + n$

$$\begin{aligned} \text{soma}(n) &= n + \text{soma}(n - 1); \\ \text{soma}(1) &= 1. \end{aligned}$$

$$\begin{aligned} \text{soma}(6) &= 6 + \text{soma}(5) \\ &= 6 + (5 + \text{soma}(4)) \\ &= 6 + (5 + (4 + \text{soma}(3))) \\ &= 6 + (5 + (4 + (3 + \text{soma}(2)))) \\ &= 6 + (5 + (4 + (3 + (2 + \text{soma}(1))))) \\ &= 6 + (5 + (4 + (3 + (2 + (1 + 1))))) = 21 \end{aligned}$$

$\text{soma}(1) = 1;$
 $\text{soma}(n) = n + \text{soma}(n - 1).$

```
int soma(int n){  
    if (n == 1) return 1;  
    return n + soma(n-1);  
}
```

Soma dos elementos de um vetor

Problema: calcular a soma dos elementos de um vetor $v[0 \dots n - 1]$
(n elementos)

Soma dos elementos de um vetor

Problema: calcular a soma dos elementos de um vetor $v[0 \dots n - 1]$ (n elementos)

```
/* algoritmo iterativo*/  
int soma(int v[], int n){  
    int s = 0, i;  
    for (i=0; i< n; i++)  
        s += v[i];  
    return s;  
}
```

Soma dos elementos de um vetor

Problema: calcular a soma dos elementos de um vetor $v[0 \dots n - 1]$ (n elementos)

```
/* algoritmo iterativo*/  
int soma(int v[], int n){  
    int s = 0, i;  
    for (i=0; i< n; i++)  
        s += v[i];  
    return s;  
}
```

```
/* algoritmo recursivo*/  
  
int soma_r(int v[], int n){  
  
    if ( n == 1) return v[0];  
  
    return v[n-1] + soma_r(v,n-1);  
}
```

Máximo Divisor Comum

$$\begin{aligned} \text{mdc}(a, b) &= \text{mdc}(b, a \bmod b); \\ \text{mdc}(a, 0) &= a. \end{aligned}$$

Máximo Divisor Comum

$$\begin{aligned} \text{mdc}(a, b) &= \text{mdc}(b, a \bmod b); \\ \text{mdc}(a, 0) &= a. \end{aligned}$$

$$\begin{aligned} \text{mdc}(642, 36) &= \text{mdc}(36, 30) \\ &= \text{mdc}(30, 6) \\ &= \text{mdc}(6, 0) \\ &= 6. \end{aligned}$$

Máximo Divisor Comum em C

$$\begin{aligned} \text{mdc}(a, 0) &= a; \\ \text{mdc}(a, b) &= \text{mdc}(b, a \bmod b). \end{aligned}$$

```
/* Função para calcular o mdc de a e b: a>=b >=0 */
```

```
int mdc(int a, int b){  
  
    if (b == 0) return a;  
  
    return mdc(b, a % b);  
  
}
```

Sequência de Fibonacci

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55,...

Sequência de Fibonacci

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55,...

$\text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2)$

$\text{Fib}(0) = 0$

$\text{Fib}(1) = 1$

Sequência de Fibonacci

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55,...

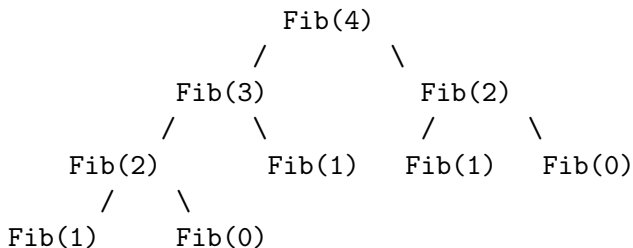
$\text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2)$

$\text{Fib}(0) = 0$

$\text{Fib}(1) = 1$

$\text{Fib}(4) = \text{Fib}(3)$	$+ \text{Fib}(2)$
$\text{Fib}(2) + \text{Fib}(1)$	$+ \text{Fib}(2)$
$\text{Fib}(1) + \text{Fib}(0) +$	$\text{Fib}(1) + \text{Fib}(2)$
$1 + 0 +$	$\text{Fib}(1) + \text{Fib}(2)$
$1 +$	$1 + \text{Fib}(2)$
$2 + \text{Fib}(2)$	
$2 + \text{Fib}(1) + \text{Fib}(0)$	
$2 + 1 + \text{Fib}(0)$	
$3 + \text{Fib}(0)$	
$3 + 0$	
3	

Calcular Fib(4):



Fibonacci recursivo em C:

$$\textit{Fibonacci}(0) = 1;$$

$$\textit{Fibonacci}(1) = 1;$$

$$\textit{Fibonacci}(n) = \textit{Fibonacci}(n - 1) + \textit{Fibonacci}(n - 2).$$

Fibonacci recursivo em C:

$Fibonacci(0) = 1;$
 $Fibonacci(1) = 1;$
 $Fibonacci(n) = Fibonacci(n - 1) + Fibonacci(n - 2).$

```
int fibonacci(int n) {  
  
    if (n == 0 || n==1) return 1;  
  
    return fibonacci(n-1) + fibonacci(n-2);  
}
```

Fibonacci iterativo em C:

```
int fibonacci_iterativo(int n) {  
  
    int f0 = 0, f1 = 1, f2, i;  
    if (n == 0 || n == 1) return 1;  
  
    for(i=0; i<= n-2; i++){  
        f2 = f1 + f0;  
        f0 = f1;  
        f1 = f2;  
    }  
    return f2;  
}
```


Exponenciação

Problema: calcular $a^n = \underbrace{a \cdot a \cdot a \cdots a}_{n \text{ vezes}}$

Exponenciação

Problema: calcular $a^n = \underbrace{a \cdot a \cdot a \cdots a}_{n \text{ vezes}}$

```
/* algoritmo iterativo*/
```

```
int exp_it(int a, int n){  
  
    int p = 1, i;  
  
    if (n == 0) return 1;  
  
    for (i=1; i<= n; i++)  
        p = p * a;  
  
    return p;  
}
```

$$a^n = \begin{cases} 1 & \text{se } n = 0; \\ a \cdot a^{n-1} & \text{se } n > 0. \end{cases}$$

$$a^n = \begin{cases} 1 & \text{se } n = 0; \\ a \cdot a^{n-1} & \text{se } n > 0. \end{cases}$$

```
/* algoritmo recursivo 1:*/
```

```
int exp_r(int a, int n){  
    if ( n==0) return 1;  
  
    return a * exp_r(a,n-1);  
}
```

Exponenciação-Recursiva II

$$a^n = \begin{cases} a \cdot (a^{\lfloor n/2 \rfloor})^2 & \text{se } n \text{ ímpar;} \\ (a^{\lfloor n/2 \rfloor})^2 & \text{se } n \text{ par.} \end{cases}$$

Exemplo: a^{24}

$$a^{24} = \boxed{a^{12}}^2 = \dots = \boxed{\boxed{\boxed{a \cdot \boxed{a^2}^2}^2}^2}^2$$

4 quadrados + 1 multiplicação (melhor que 23 multiplicações!)

$$a^n = \begin{cases} a \cdot (a^{\lfloor n/2 \rfloor})^2 & \text{se } n \text{ ímpar;} \\ (a^{\lfloor n/2 \rfloor})^2 & \text{se } n \text{ par.} \end{cases}$$

```
/* algoritmo recursivo: 2*/
```

```
int exp_r2(int a, int n){
```

```
    int b;
```

```
    if ( n==0) return 1;
```

```
    b = exp_r2(a,n/2);
```

```
    if (n % 2 == 1) return a*b*b;
```

```
    else return b*b;
```

Máximo de um vetor:

Problema: encontrar o valor do maior elemento do vetor $v[0 \dots n - 1]$ (n elementos)

Máximo de um vetor:

Problema: encontrar o valor do maior elemento do vetor

$v[0 \dots n - 1]$ (n elementos)

```
int maior(int v[], int n){ /* algoritmo iterativo: */  
    int x = v[0], i;  
    for (i=1; i< n; i++)  
        if (x > v[i]) x = v[i];  
    return x;  
}
```


Máximo de um vetor:

Problema: encontrar o valor do maior elemento do vetor

$v[0 \dots n - 1]$ (n elementos)

```
int maior(int v[], int n){ /* algoritmo iterativo: */  
    int x = v[0], i;  
    for (i=1; i< n; i++)  
        if (x > v[i]) x = v[i];  
    return x;  
}
```

```
int maior_r(int v[], int n){ /* algoritmo recursivo: */  
    int x;  
    if ( n==1) return v[0];  
    x = maior_r(v,n-1);  
    if (x > v[n-1]) return x;  
    else return v[n-1];  
}
```