

Universidade Federal da Paraíba

Centro de Informática

Departamento de Informática

Estrutura de Dados

Recursividade

▶ Tiago Maritan

▶ tiago@ci.ufpb.br

Recursividade – Motivação

- ▶ Sabemos que uma função pode chamar outras funções
 - ▶ Ex: **func1 ()** chamando a **func2 ()**

```
void func2 () {  
    printf("Na função 2");  
}  
  
void func1 () {  
    printf("Na função1");  
    printf("Vou chamar a função2");  
    func2 () ;  
    printf("No fim da função1");  
}
```

Recursividade - Motivação

- ▶ Hoje veremos que uma função pode chamar ela mesma

- ▶ Ex: `func1()` chamando a si mesma

```
void func1() {  
    printf("No inicio da função1");  
    func1() ;  
    printf("No fim da função1");  
}
```

- ▶ Mas isso é útil???

- ▶ No exemplo acima, não! Chama `func1()` infinitamente!

- ▶ E existe algum caso onde isso possa ser útil?

- ▶ Existe sim! Nas **funções recursivas**! Estudaremos isso na aula de hoje!

Função Recursiva

- ▶ Função que chama a si mesma
- ▶ Deve conter duas partes:
 - ▶ **Caso recursivo** – no qual a **função chama a si mesma**
 - ▶ **Caso não-recursivo** – estabelece uma **condição de parada** da recursão
- ▶ Deve-se garantir que uma chamada recursiva atinja, em algum momento, a **condição de parada**

Função Recursiva

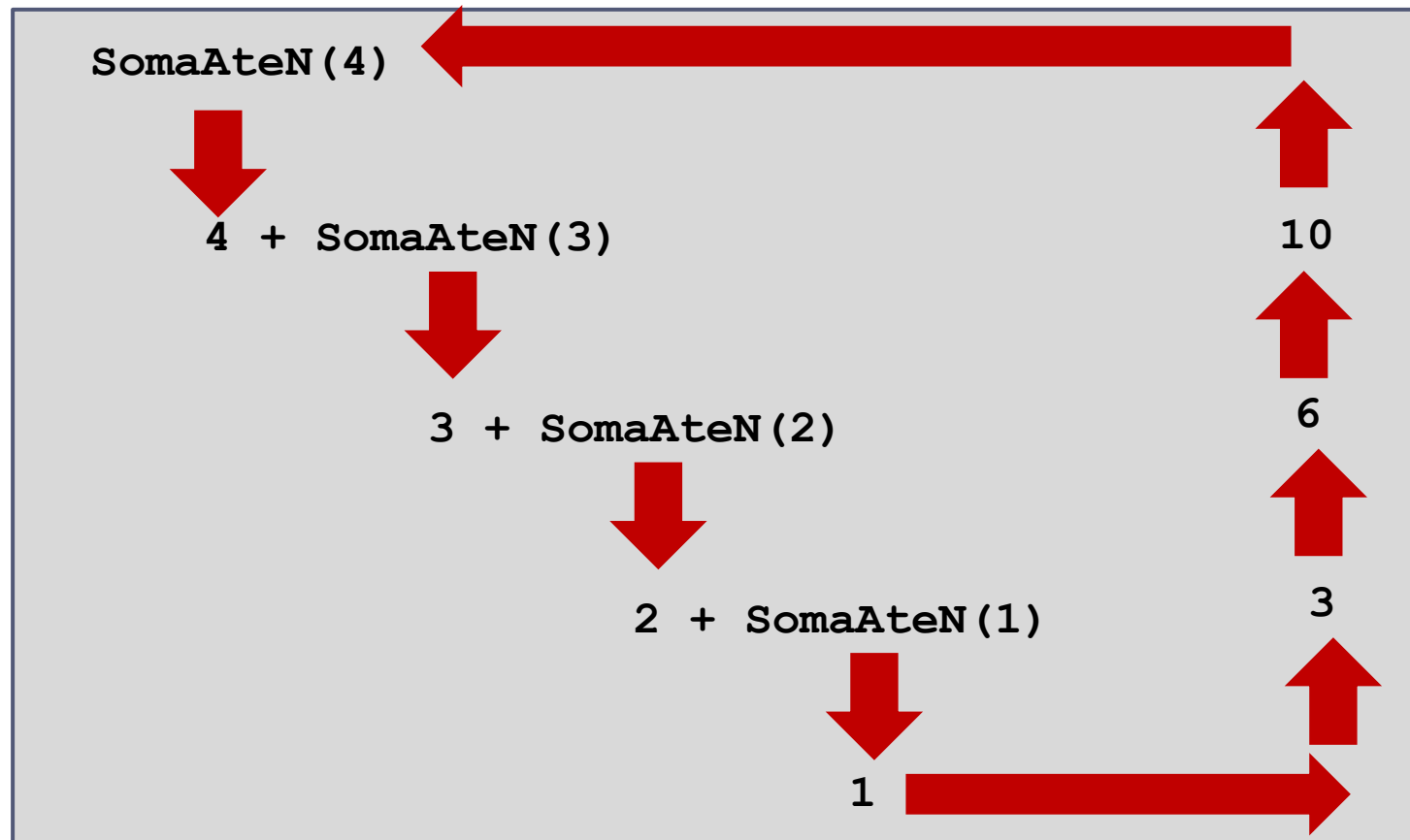
- ▶ Exemplo I: Função que retorna a soma de inteiros de 1 e n

```
//Retorna a soma dos inteiros entre 1 e n
int SomaAteN(int n){
    if (n <= 1){
        // Condição de parada
        return 1;
    }
    else{
        // Caso recursivo
        return (n + SomaAteN(n-1));
    }
}
```

- ▶ Como funciona???

Função Recursiva

- ▶ Exemplo I: Função que retorna a soma de inteiros de 1 e n
 - ▶ Como funciona a chamada `SomaAteN(4)`?



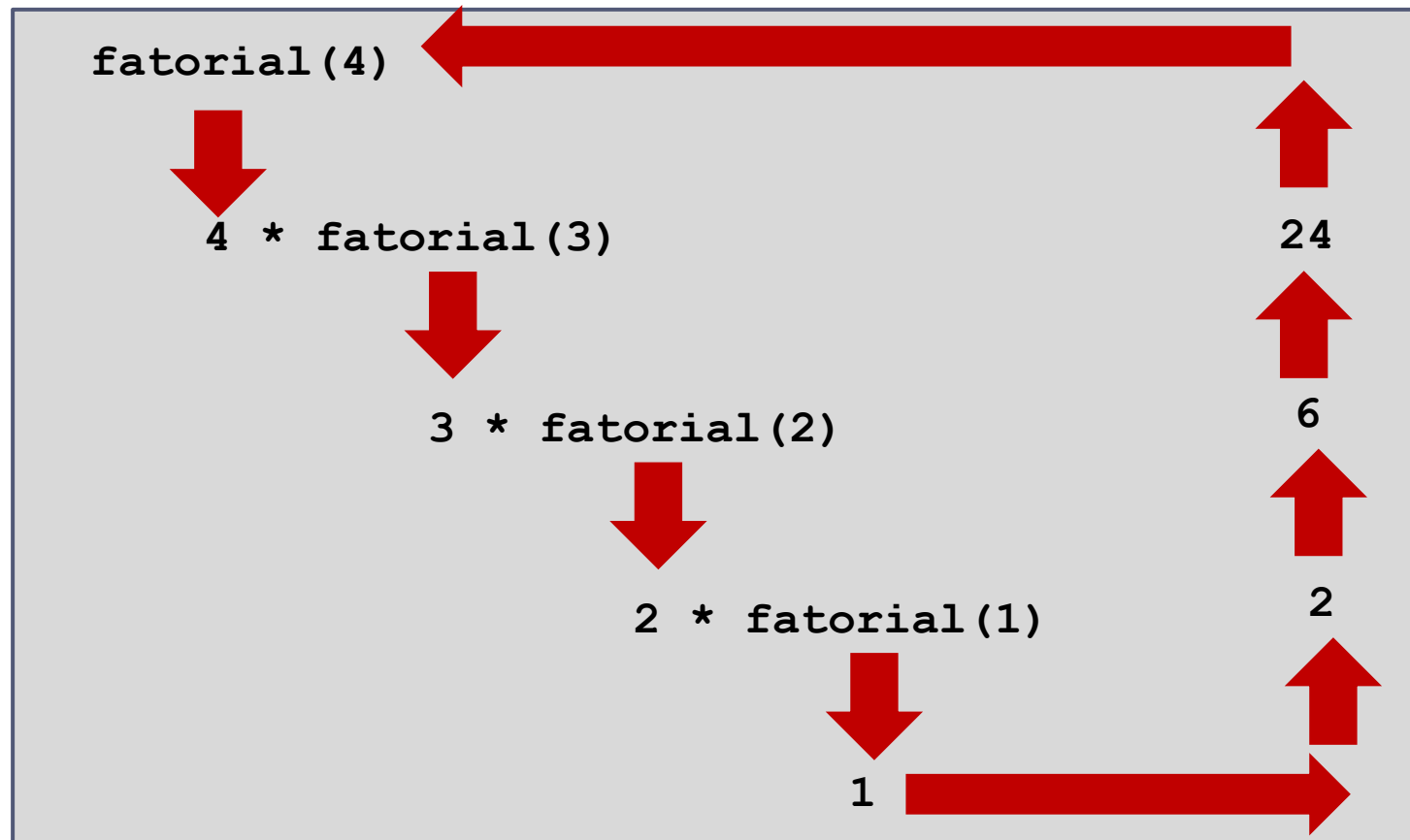
Função Recursiva

► Exemplo 2: Função fatorial

```
// Retorna o fatorial de um numero n
int fatorial (int n){
    if (n <= 1){
        // Condição de parada
        return 1;
    }
    else{
        // Caso recursivo
        return (n * fatorial(n - 1));
    }
}
```

Função Recursiva

- ▶ Exemplo I: Função que retorna a soma de inteiros de 1 e n
 - ▶ Como funciona a chamada `fatorial(4)`?



Quando usar recursividade?

- ▶ Quando o problema tem uma **estrutura recursiva**
 - ▶ Ou seja, uma **instância** do problema contém uma **instância menor** do mesmo problema.
 - ▶ Ex: `fatorial(n) = n*fatorial(n-1)`
 - ▶
 - ▶ Menor instância pode ser resolvida diretamente;
 - ▶ Ex: `fatorial(1) = 1;`
- ▶ Quando a solução iterativa do problema (usando *while*, *for*, etc.) é complexa.
 - ▶ Ex: Torres de Hanoi;

Observações sobre Recursividade

- ▶ É uma técnica elegante e, geralmente, quem a utiliza demonstra experiência,
- ▶ Mas o seu uso possui um **preço**:
 - ▶ Movimentação de dados na PILHA de chamadas de funções.
 - ▶ Execução, geralmente, mais lenta que as soluções iterativas;.
- ▶ Portanto, deve se dar preferência para soluções iterativas
 - ▶ Utilizar recursividade apenas nos casos apropriados
 - ▶ Atentar para as características básicas de um problema recursivo