

MINIAULA DE ALGORITMOS SUBRPOGRAMAÇÃO

Prof. Ivanilton Polato

Departamento Acadêmico de Computação (DACOM-CM)

ipolato@utfpr.edu.br

Subprogramação: o que é?

- Mecanismo de organizar seu código!
- Maneira de lidar com o aumento da complexidade
- Realizado através da criação de funções
- Tarefas repetidas ao longo do código são organizadas e centralizadas em um ponto

Funções: o que são?

- As funções são trechos de código com um objetivo específico
- Podem ser chamadas ao longo do programa, conforme a necessidade
- Como desenvolver:
 - *Procurar por trechos do algoritmo com ações específicas*
 - *Isolar e atribuir um nome para esse trecho*
 - Definição das informações recebidas/devolvidas
 - *Utilizar esse trecho de código através da função gerada*

Funções: exemplificando

- Observe o algoritmo ao lado
- Uma mesma solução (X) desenvolvida foi repetida em dois pontos do algoritmo...
- E se a solução estivesse presente em muitos pontos do algoritmo?
- Como fazer e controlar modificações uniformes?

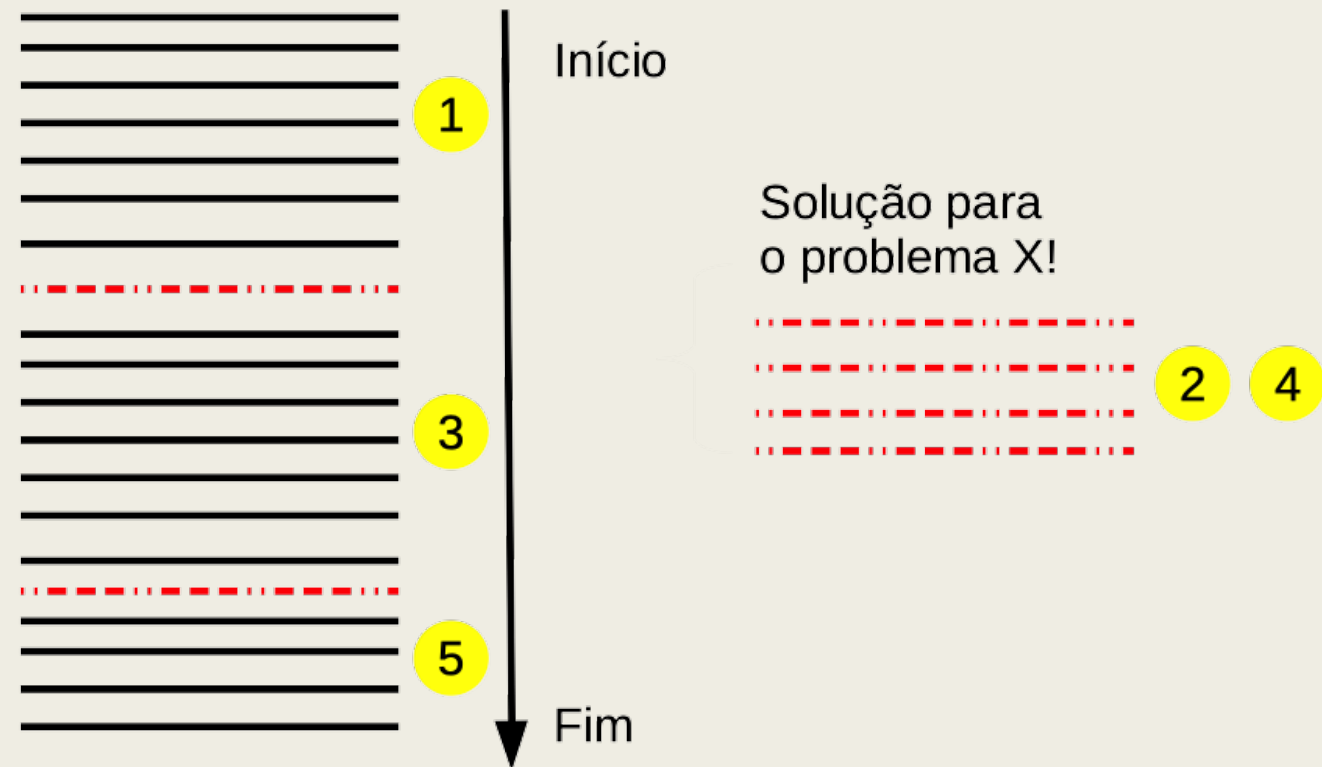
Algoritmo sem subprogramação (funções)



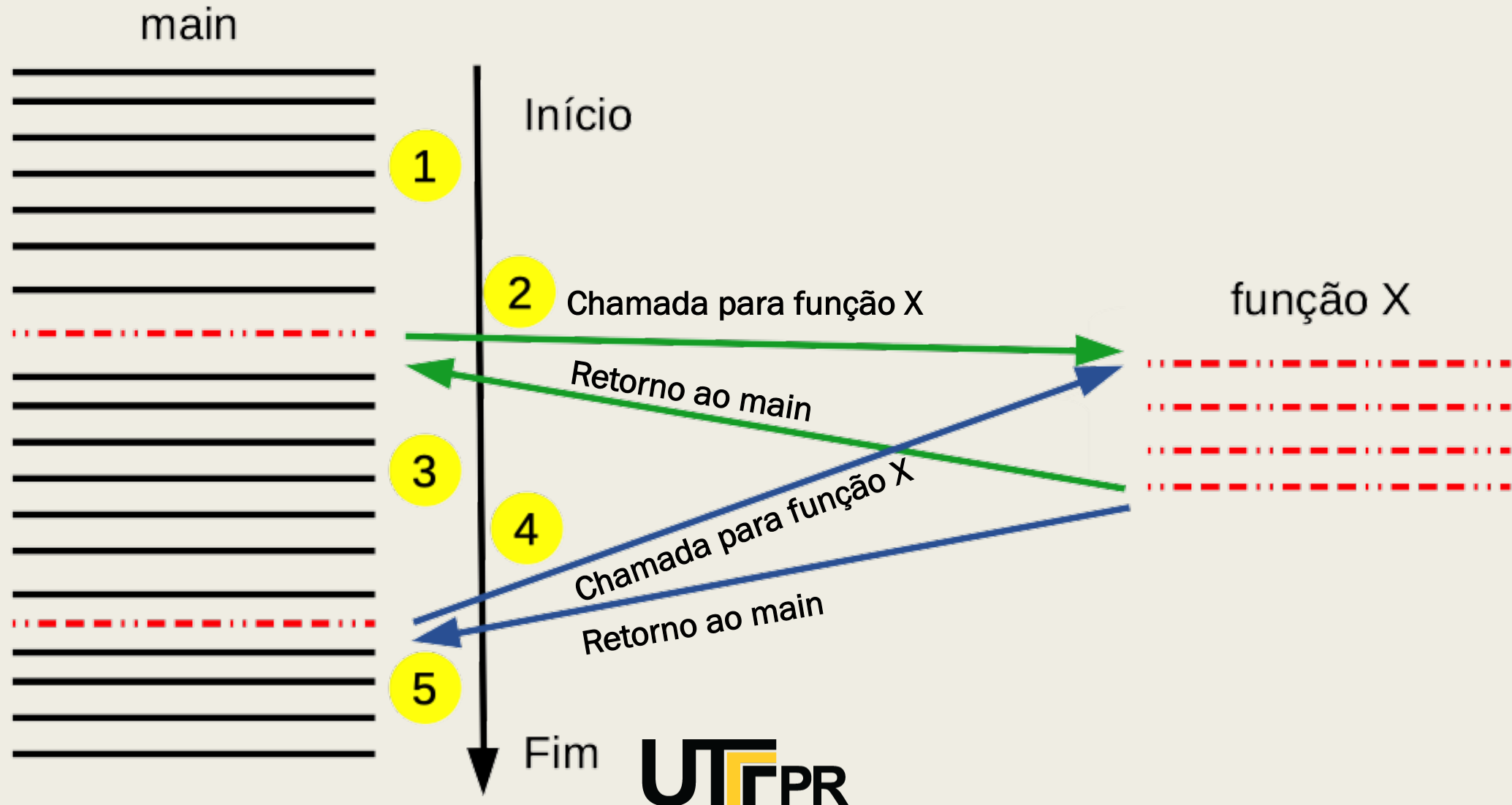
Funções: exemplificando

- Identificamos que a solução de X pode ser uma função!
- Isolamos e atribuímos um nome para esse trecho de código
- Substituímos esse código pela chamada da nova função

Algoritmo com subprogramação (funções)



Funções: exemplificando



Escopo de variáveis

- É o conceito que define a visibilidade e contexto das variáveis!
- **Variáveis locais:** são visíveis dentro de uma função onde foram declaradas:
 - *Só conseguem ser utilizadas dentro dessa função*
- **Variáveis globais:** são declaradas fora das funções e visíveis por qualquer função do seu programa:
 - *Podem ser utilizadas em qualquer função dentro de seu programa*

Passagem de parâmetros e Retorno

- Para permitir a comunicação entre funções existem esses dois conceitos
- Passagem de parâmetros: permite que a função receba valores enviados pela instrução que está chamando a função quando esta for iniciar sua execução
- Retorno: permite a devolução de um valor calculado pela função para a instrução que a chamou

Passagem de parâmetros e Retorno

- Exemplo: Um programa possui uma função que soma dois números. Nesse caso, quando a função for utilizada o programa deve enviar dois números e receber o resultado da soma.
- **Passagem de parâmetros**, no exemplo, é o envio dos para que a função faça a soma deles
- **Retorno**, no exemplo, é a devolução do resultado da soma dos números para o programa principal

Funções, parâmetros e retorno

- O uso de parâmetros e retorno é opcional!
- Uma função pode ter diversos parâmetros!
- A princípio, uma função pode devolver um único valor!
- O desenvolvedor pode criar quantas funções forem necessárias!
- Estudaremos quatro tipos de funções:
 1. *Sem passagem de parâmetros e sem retorno;*
 2. *Com passagem de parâmetros e sem retorno;*
 3. *Sem passagem de parâmetros e com retorno;*
 4. *COM passagem de parâmetros e COM retorno!*