Algoritmo e Estrutura de Dados II
COM-112

Ordenação por Seleção
Selection Sort

Vanessa Souza

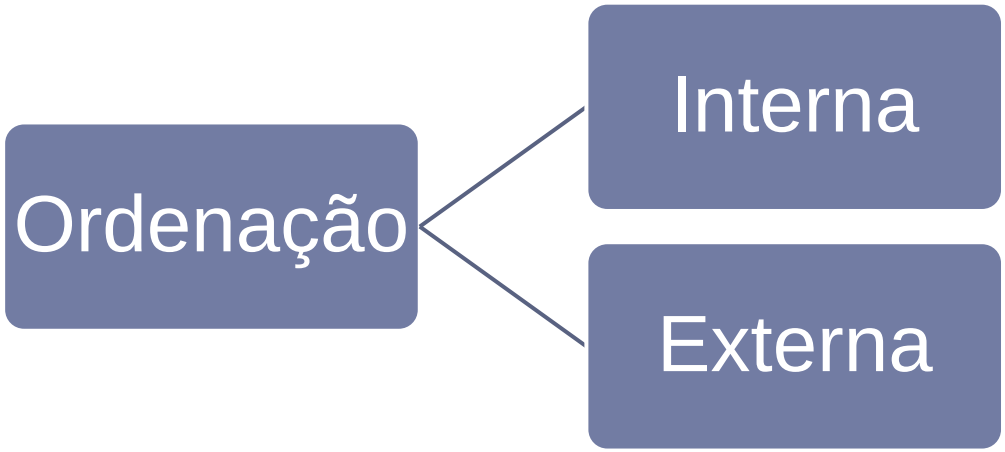
Universidade Federal de Itajubá – Instituto de Matemática e Computação

Ordenação

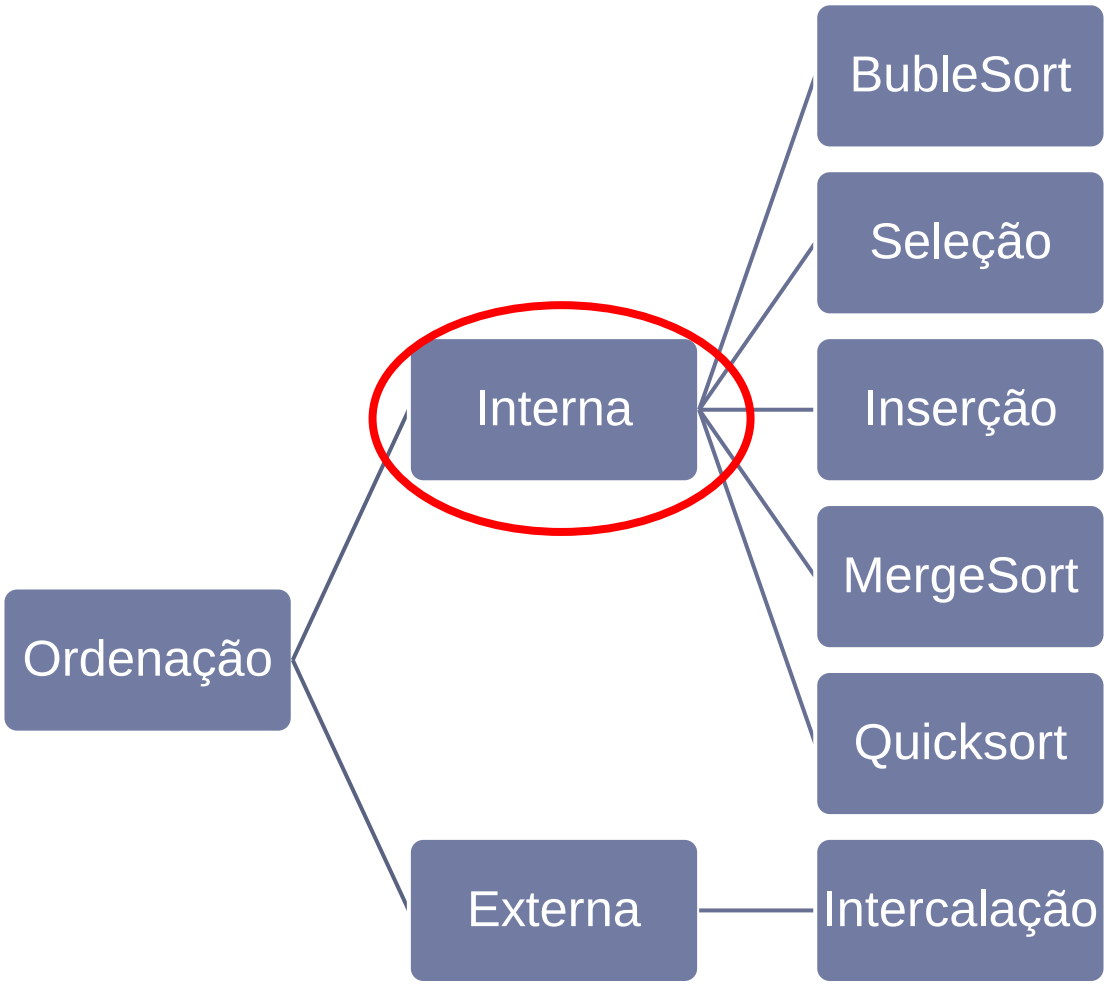
Ordenação

- Processo bastante utilizado na computação de uma estrutura de dados
- Ordenar significa colocar em ordem, segundo algum critério
- Alterar a ordem na qual os elementos de uma estrutura de dados aparece nessa estrutura
  - Rearranjar a estrutura

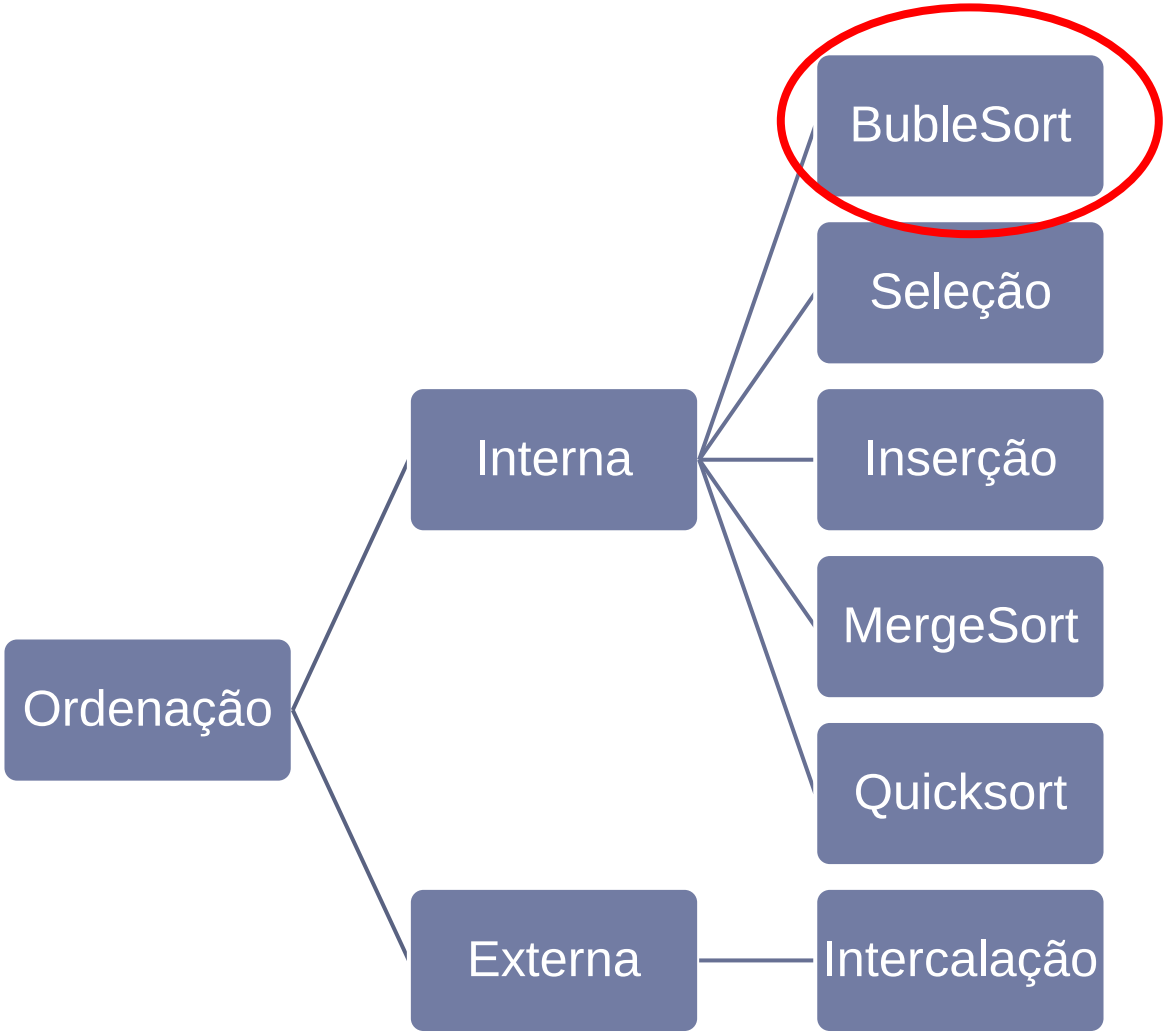
Classificação dos Métodos de Ordenação



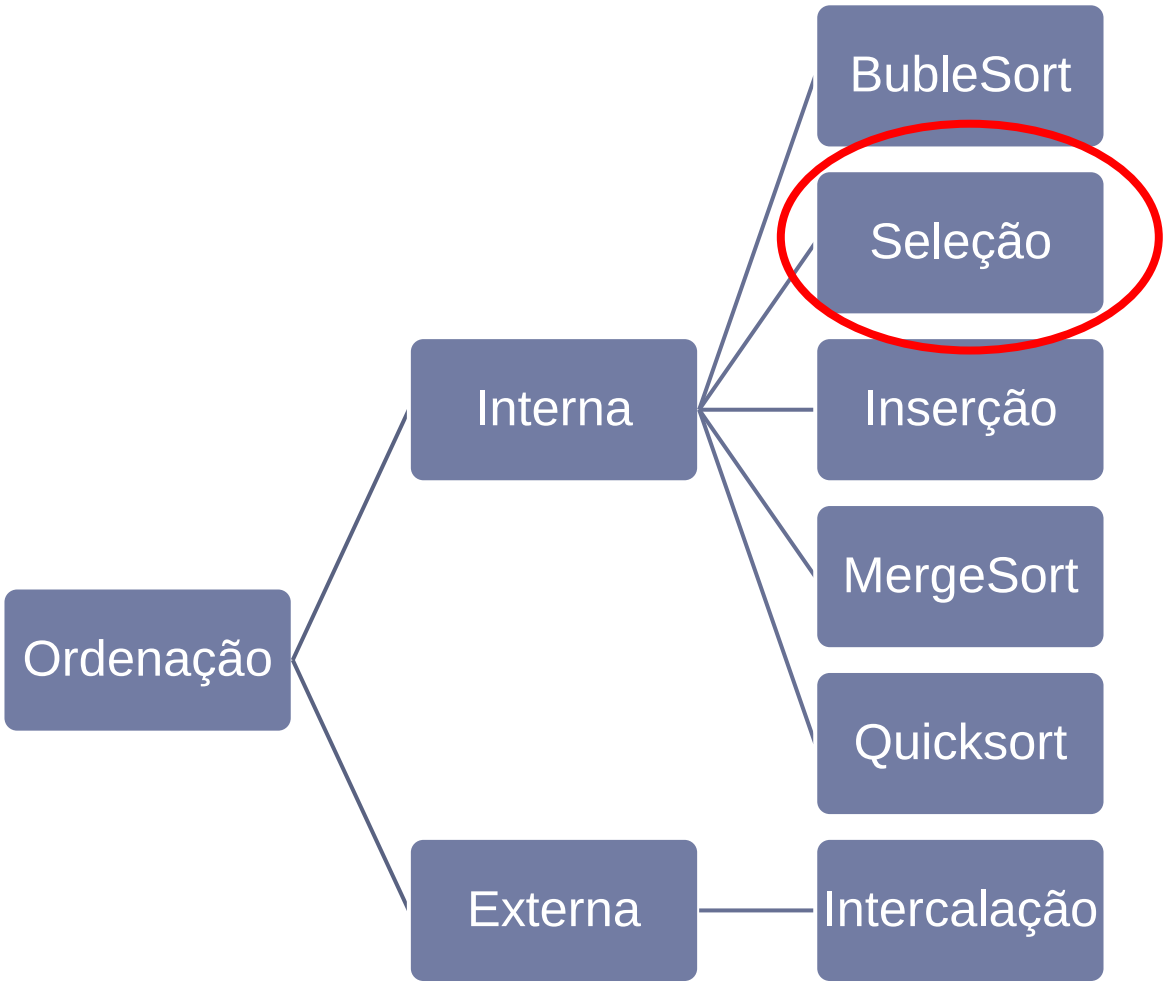
Classificação dos Métodos de Ordenação



Classificação dos Métodos de Ordenação



Classificação dos Métodos de Ordenação

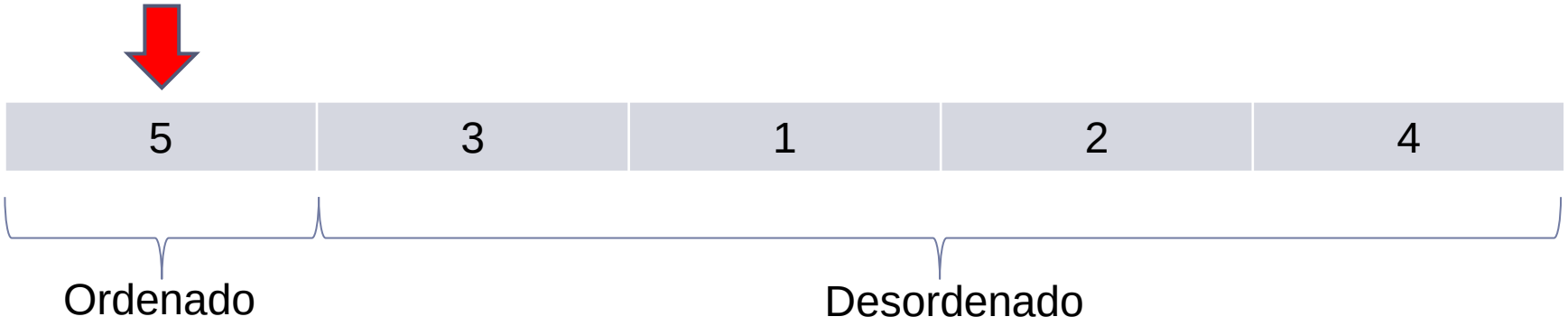


Seleção

Ordenação por Seleção

- Ideia
  - Este algoritmo usa um marcador para dividir as partes ordenada (à esquerda) e desordenada (à direita) do vetor.
  - Procura-se na parte desordenada pelo menor elemento e troca-se esse elemento com o elemento sob o marcador.
  - Em seguida, avança-se o marcador. O processo se repete até que exista apenas um elemento a partir do marcador.

Ordenação por Seleção



Ordenação por Seleção

Procura pelo menor elemento

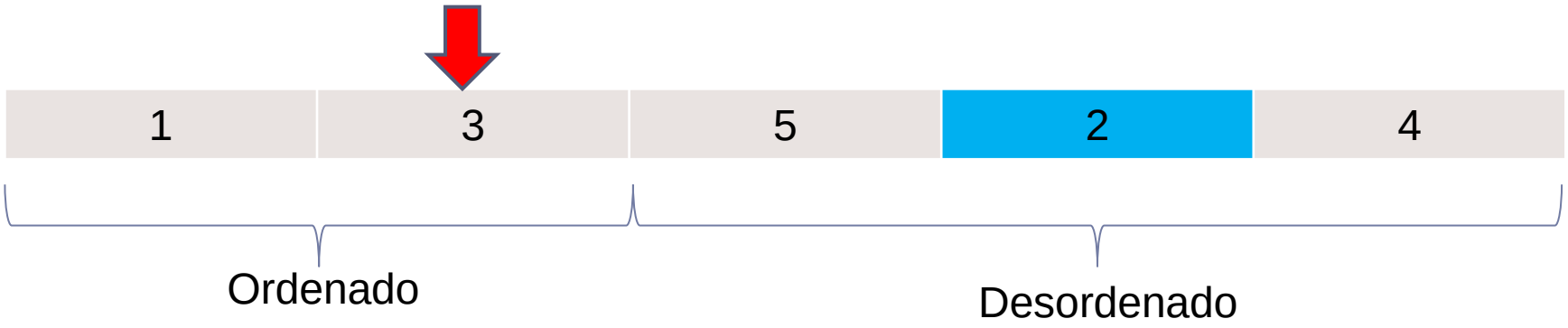
Ordenação por Seleção

Troca com a posição do marcador

Ordenação por Seleção

Anda com o marcador

Ordenação por Seleção

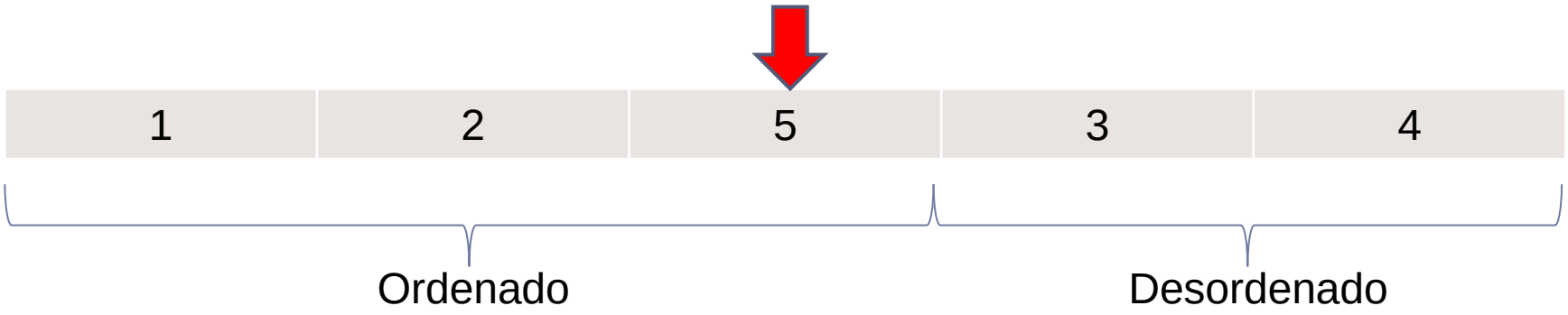


Repete o processo

Ordenação por Seleção

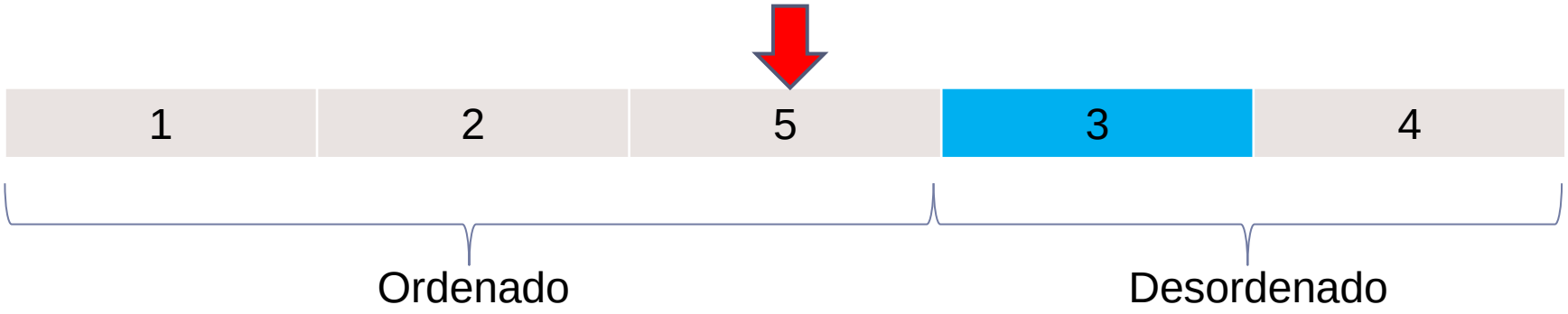
Repete o processo

Ordenação por Seleção

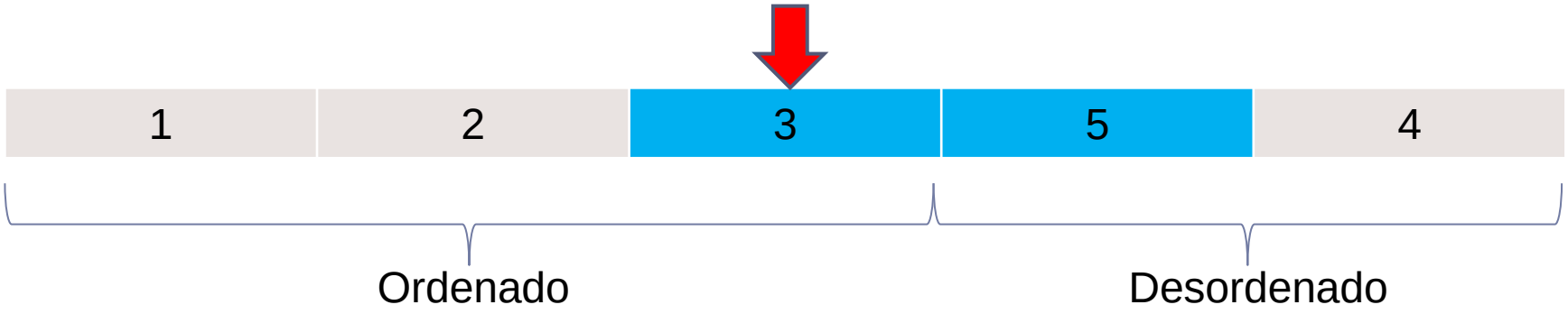


Repete o processo

Ordenação por Seleção

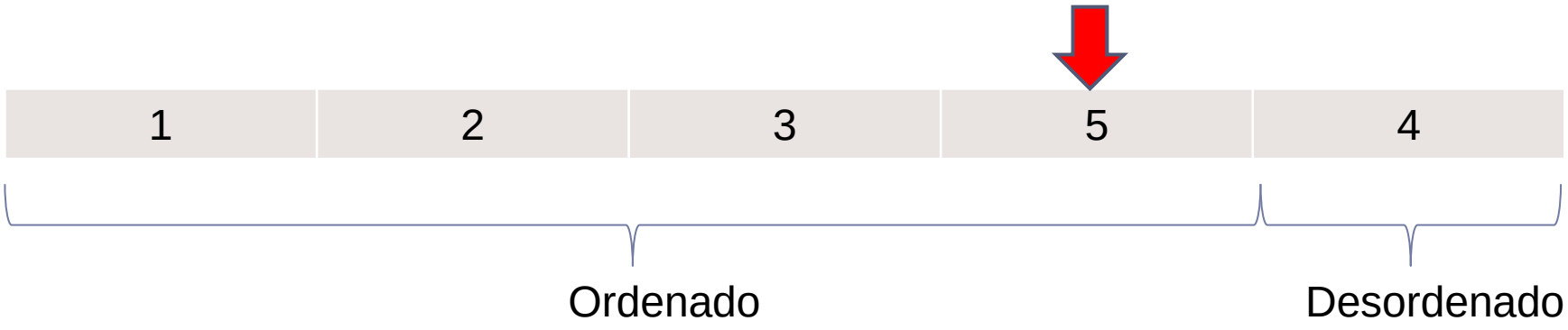


Ordenação por Seleção



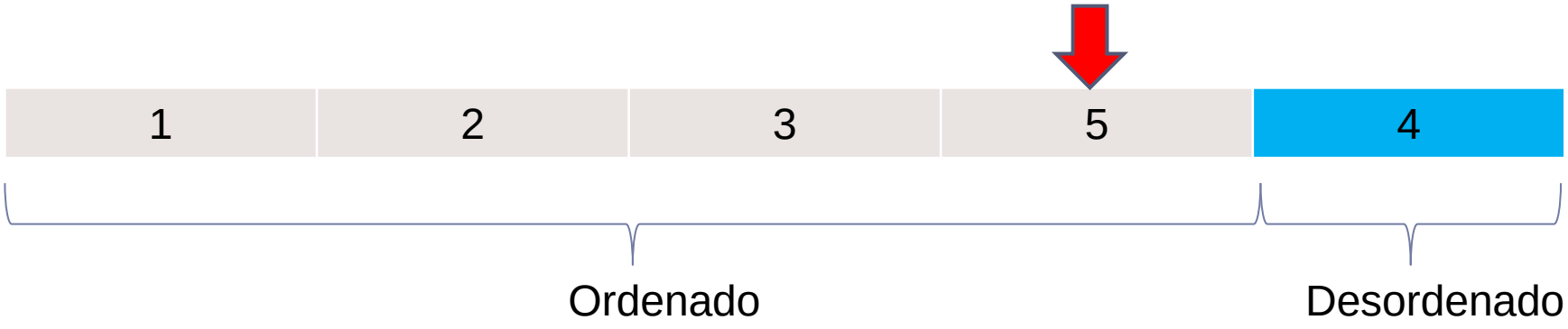
Repete o processo

Ordenação por Seleção



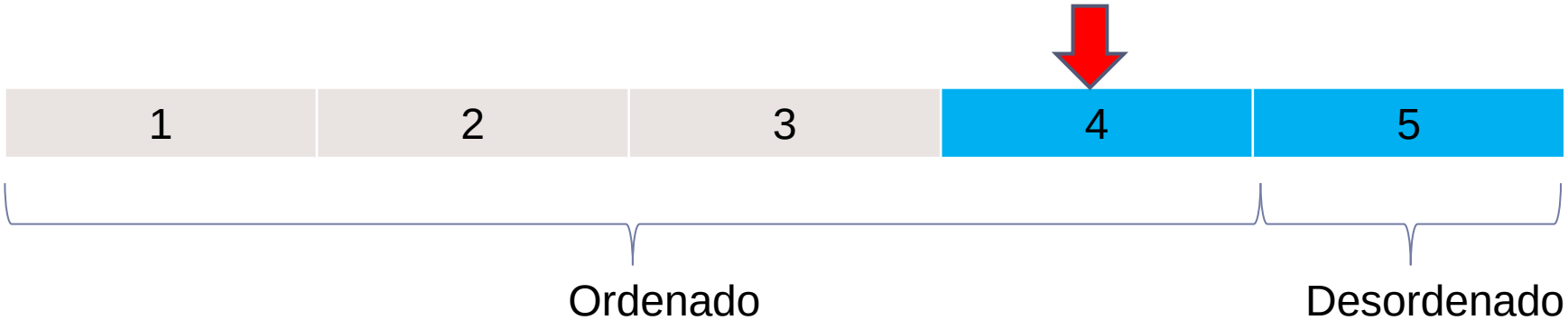
Repete o processo

Ordenação por Seleção



Repete o processo

Ordenação por Seleção

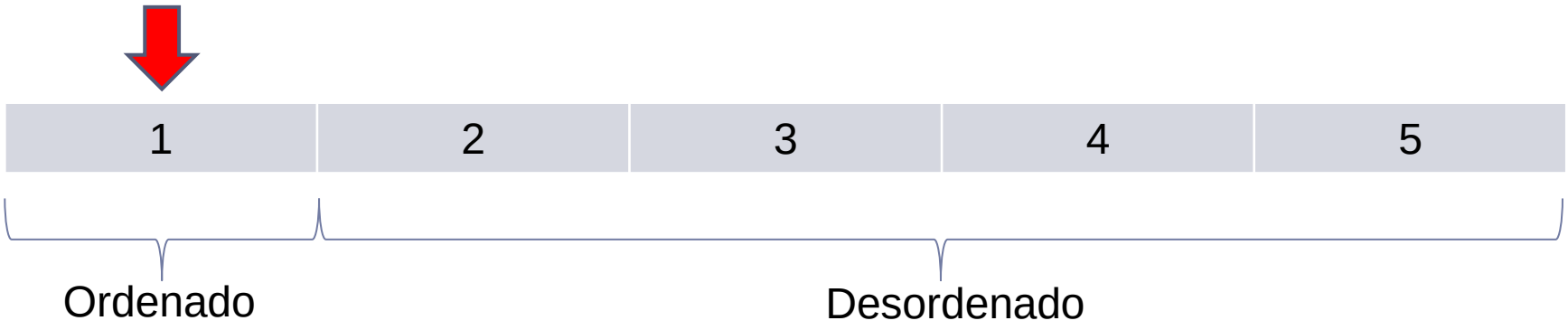


Repete o processo

Ordenação por Seleção

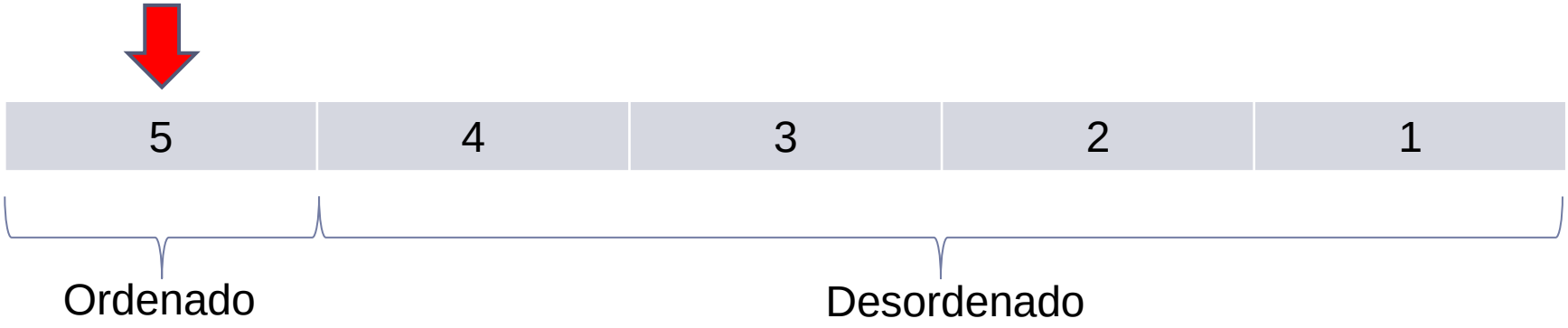
Ordenação por Seleção

- Exercício
  - Ordenar por seleção o seguinte vetor



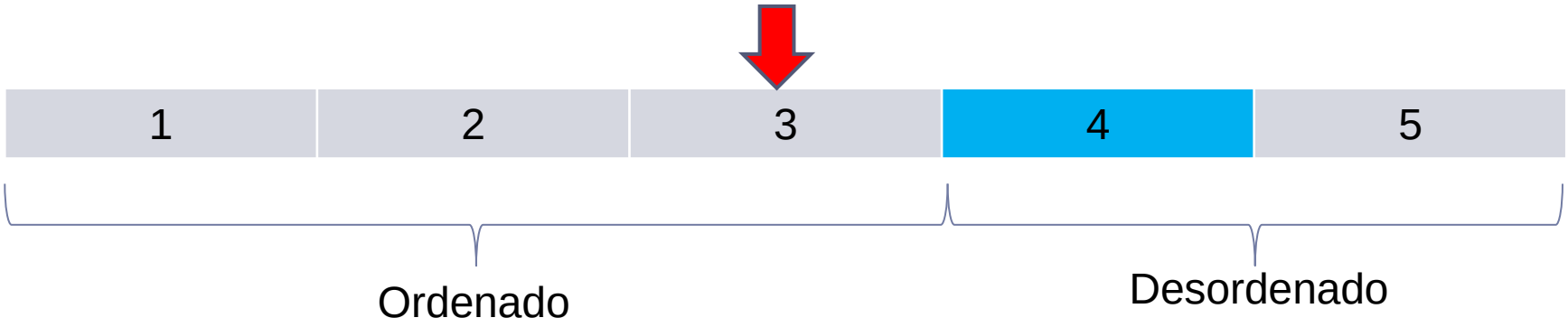
Ordenação por Seleção

- Exercício
  - Ordenar por seleção o seguinte vetor



Ordenação por Seleção

- Exercício
  - Ordenar por seleção o seguinte vetor



Ordenação por Seleção

- Vamos programar??

Algorithm 1 Ordenação por Seleção

```
procedure SELECAO(V, tamVet)                    ▷ V é um vetor numérico
    marcador ← 0
    menor ← 0
    while (marcador < tamVet − 1) do
        menor ← índice do menor elemento da parte desordenada do vetor
        if (vet[menor] < vet[marcador]) then
            troque vet[marcador] com vet[menor]
        end if
        marcador ← marcador + 1
    end while
end procedure
```

Ordenação por Seleção

- Vamos programar??

Algorithm 1 Ordenação por Seleção

```
procedure SELECAO(V, tamVet)                    ▷ V é um vetor numérico
    marcador ← 0
    menor ← 0
    while (marcador < tamVet − 1) do
        menor ← índice do menor elemento da parte desordenada do vetor
        if (vet[menor] < vet[marcador]) then
            troque vet[marcador] com vet[menor]
        end if
        marcador ← marcador + 1
    end while
end procedure
```

Modularização

- Decompor programas complexos em programas menores e depois juntá-los para compor o programa final.
  - Facilita a implementação
  - Facilita os testes
  - Permite reuso
  - Permite divisão de tarefas entre os codificadores

Modularização

- Na linguagem C:
  - A modularização começa através do uso adequado de funções
    - Tipicamente, usamos funções para realizar tarefas que se repetem várias vezes na execução de um mesmo programa.
  - Na linguagem C, cada módulo é chamado de função.
  - Um problema acontece quando existe uma grande quantidade de funções, com objetivos diferentes, escritas num mesmo código fonte.
    - Neste caso precisamos criar arquivos de código-fonte diferentes.

Modularização

- Na linguagem C:
  - Exemplo prático são as bibliotecas-padrão de C
    - stdio
    - math
    - String
  - Quando programas de grande dimensão são implememententados, estes devem ser divididos em módulos.
    - Estes módulos, em C, são simplesmente arquivos contendo coleções de funções, de algum modo **relacionadas entre si**.

Modularização

- Biblioteca x Arquivos-Objetos
  - Ambos são coleções de funções
  - Include em uma biblioteca carrega para o seu programa apenas as funções utilizadas.
  - Include em um arquivo de objetos carrega todo o conteúdo do arquivo de objetos (utilizados ou não) para o seu programa
  - *Link*

Modularização

- Uma Biblioteca da Linguagem C é composta por dois tipos de arquivos:
  - **Cabeçalho** (*header*) - tem extensão *.h*
  - **Objeto pré-compilado** (binário) - pode ter extensão *.a* (biblioteca estática) ou .so (biblioteca compartilhada) e corresponde ao executável da biblioteca (os cabeçalhos apenas declaram as funções).
    - O cabeçalho será importante para o compilador encontrar todas as referências feitas no programa C.
    - Mas na hora de gerar o arquivo executável, o compilador incluirá o objeto pré-compilado correspondente.

Modularização

- A maioria dos compiladores C inclui instruções para criar uma biblioteca.
- Uma vez que esse processo de compilador para compilador, estude seu manual do usuário para determinar que procedimento você deve seguir
- No gcc, uma forma comum de se fazer isso no linux é utilizando o aplicativo **ar**
  - http://www.adp-gmbh.ch/cpp/gcc/create_lib.html
  - http://www.network-theory.co.uk/docs/gccintro/gccintro_79.html

Modularização

Cabeçalho (*header*) - tem extensão *.h*
Arquivo de código fonte - tem extensão *.c*

Modularização

- Arquivo de Cabeçalho
  - Protótipos das funções
  - Comentários
  - Descrição dos parâmetros de entrada e saída
  - Exemplos de como usar
  - Structs
- Os arquivos .h são uma espécie de 'intermediário', entre seu programa e os módulos
- O header também é chamado de arquivo de interface de comunicação ou, mais comumente conhecido como API - *Application Programming Interface*

Modularização

- Arquivo de código fonte
  - Arquivos com código fonte da implementação

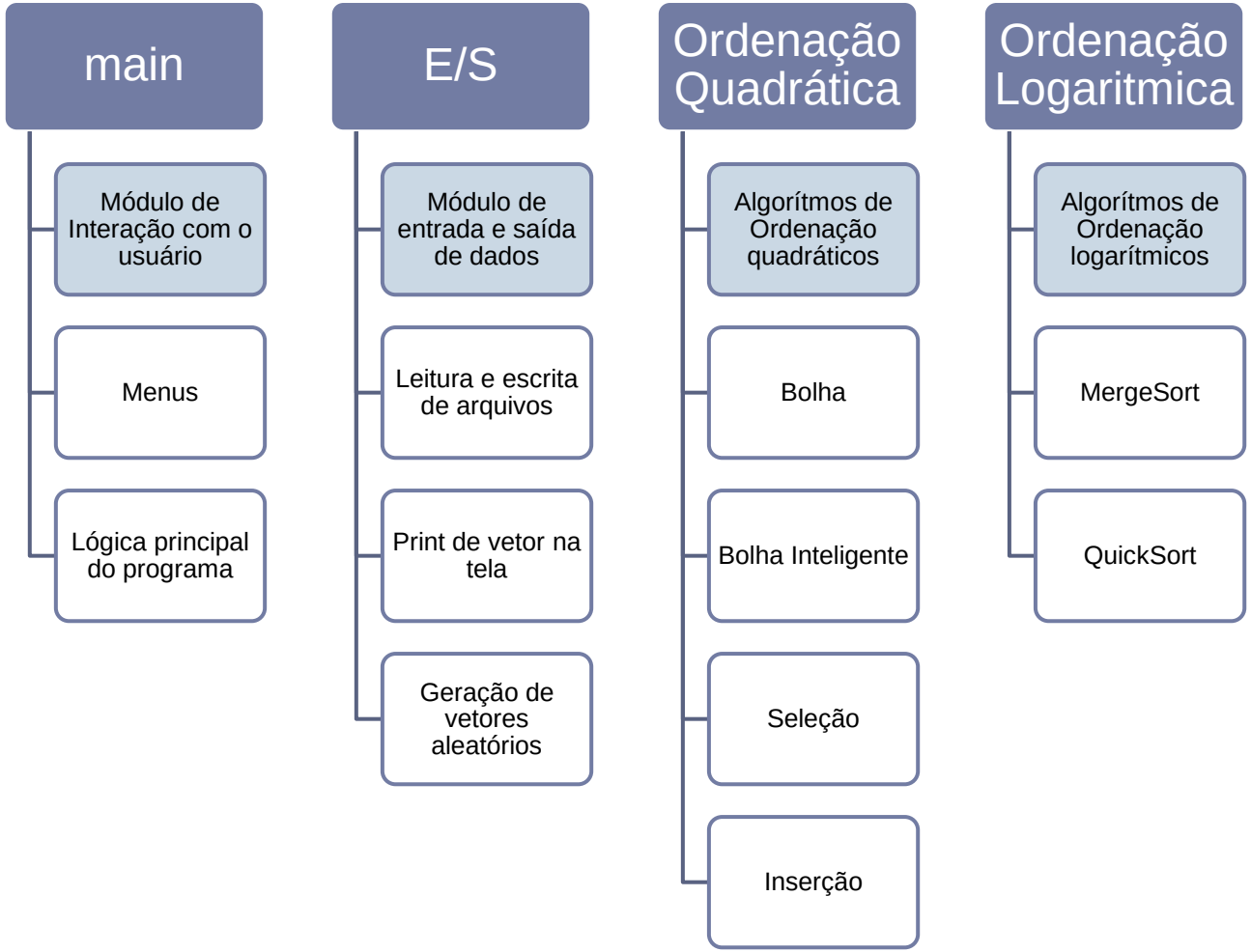
  - Os arquivos .c implementam as funções declaradas no header
  - Eles devem dar o include no arquivo header

Modularização

- Organização de projetos modularizados
  - Faça uma divisão lógica das funções
  - Nomeie os arquivos .h e .c com nomes sugestivos
  - No projeto só pode haver uma função **main**!
  - O arquivo main deve incluir os headers necessários para as chamadas das funções
  - Tudo em um arquivo só é ruim. Muitos arquivos-fonte também é ruim

- Compilação de arquivos separados
  1. Compila os arquivos de código-fonte separadamente (gera os chamados código-objeto)
  2. Linka os códigos-objetos para gerar o programa executável

Modularização – Algoritmos de Ordenação



Ordenação por Seleção

- Vamos programar??
  - Ajustar o projeto para a modelagem anterior
  - Implementar o algoritmo de ordenação por seleção

Algorithm 1 Ordenação por Seleção

```
procedure SELECAO(V, tamVet)                    ▷ V é um vetor numérico
    marcador ← 0
    menor ← 0
    while (marcador < tamVet − 1) do
        menor ← índice do menor elemento da parte desordenada do vetor
        if (vet[menor] < vet[marcador]) then
            troque vet[marcador] com vet[menor]
        end if
        marcador ← marcador + 1
    end while
end procedure
```