

# Engenharia de Computação

## **Pesquisa e Classificação de Dados**

### **Aula 03 – SelectionSort**

**Prof. Muriel de Souza Godoi**  
[muriel@utfpr.edu.br](mailto:muriel@utfpr.edu.br)

# Selection Sort

- Também conhecido como ordenação por seleção
  - É outro algoritmo de ordenação bastante simples
  - A cada passo ele seleciona o melhor elemento para ocupar aquela posição do array
    - Maior ou menor, dependendo do tipo de ordenação
    - Na prática, possui um desempenho quase sempre superior quando comparado com o BubbleSort

# Selection Sort

- Funcionamento:
  - A cada passo, procura o menor valor do array e o coloca na primeira posição
  - Divide o array em duas partes: a parte ordenada, a esquerda do elemento analisado, e a parte que ainda não foi ordenada, a direita do elemento.
- Descarta-se a primeira posição do array e repete-se o processo para a segunda posição
- Isso é feito para todas as posições do array

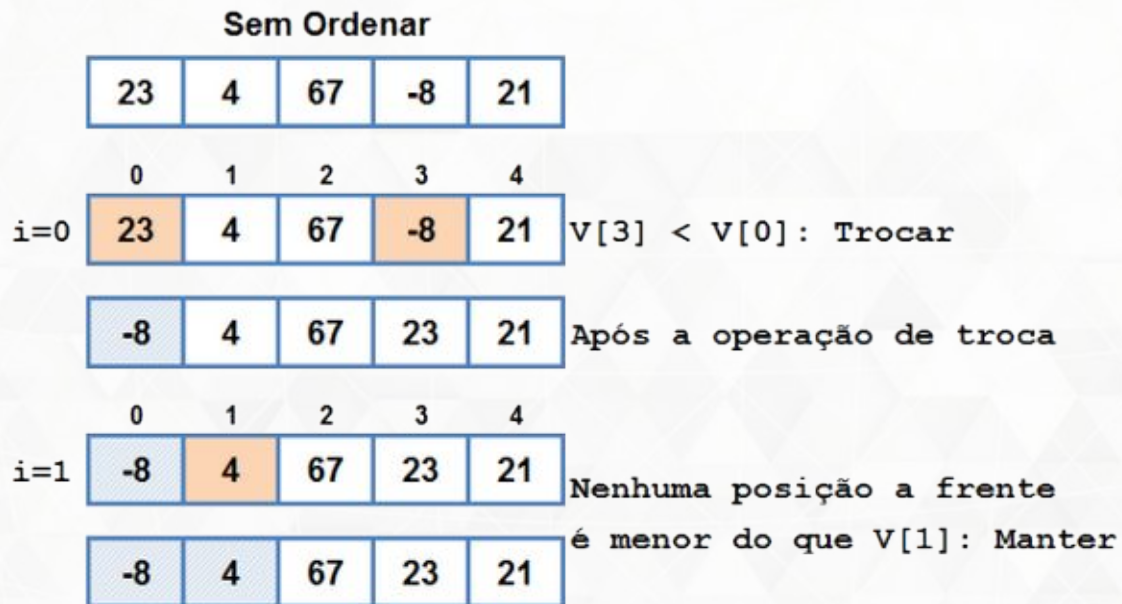
# SelectionSort - Pseudocódigo

- SelectionSort ( $V$ , tam)
  - Para cada posição  $n$  entre  $0$  e  $\text{tam}-1$  faça:
    - Encontre o índice  $i$  que contém o menor valor entre  $n$  e  $\text{tam}-1$
    - Se o menor valor é diferente do valor da posição  $n$ 
      - Troque os valores de  $V[\text{menor}]$  e  $V[n]$



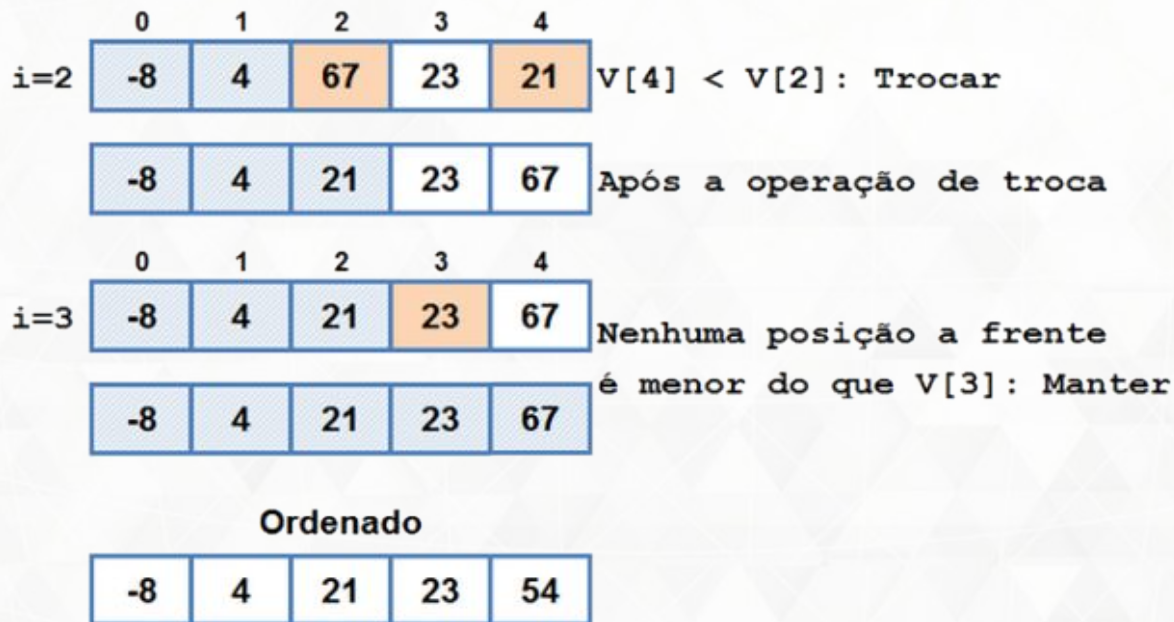
# Selection Sort

- Passo a passo:
  - Para cada posição  $i$ , procura no restante do array o menor valor para ocupá-la



# Selection Sort

- Passo a passo
  - Para cada posição  $i$ , procura no restante do array o menor valor para ocupá-la



# Selection Sort

- Vantagens

- Estável: não altera a ordem dos dados iguais

- Desvantagens

- Sua eficiência diminui drasticamente a medida que o número de elementos no array aumenta
- Não é recomendado para aplicações que envolvam grandes quantidade de dados ou que precisem de velocidade

# Selection Sort - Complexidade

- Considerando um array com N elementos, o tempo de execução é sempre de ordem  **$O(n^2)$** 
  - A eficiência do SelectionSort não depende da ordem inicial dos elementos
- Melhor do que o BubbleSort
  - Apesar de possuírem a mesma complexidade no caso médio, na prática o SelectionSort quase sempre supera o desempenho do Bubblesort
    - Envolve um número menor de comparações



# Exercício

- **1)** Simule o algoritmo utilizando cartas de um baralho ou uno.
- **2)** Implemente o **SelectionSort** em C considerando a seguinte assinatura de função

```
/**  
 * \brief Ordena o vetor usando SelectionSort  
 *  
 * \param v vetor a ser ordenado  
 * \param n tamanho do vetor  
 *  
 * Ordena o vetor usando o método SelectionSort  
 * Esse algoritmo tem um comportamento assintótico  $O(n^2)$   
 */  
void selectionSort(int *v, int n);
```