



**Conhecimento** *é a nossa natureza*

# **Estrutura de Dados I**

## **Algoritmos de Ordenação - Sort**

### **Parte 1**

Prof. Leonardo Costella



# Algoritmos de Ordenação

- É a operação de reorganizar dos dados em uma determinada ordem.
- A eficiência no manuseio de dados pode ser aumentada se os dados estiverem dispostos de acordo com algum critério de ordem
  - Exemplo:
    - Lista Telefônica;
    - Lista de Alunos (Chamada);
    - ....
  - Nesta disciplina veremos os algoritmos: ***bubblesort, selectionsort, insertionsort, quicksort e mergesort.***

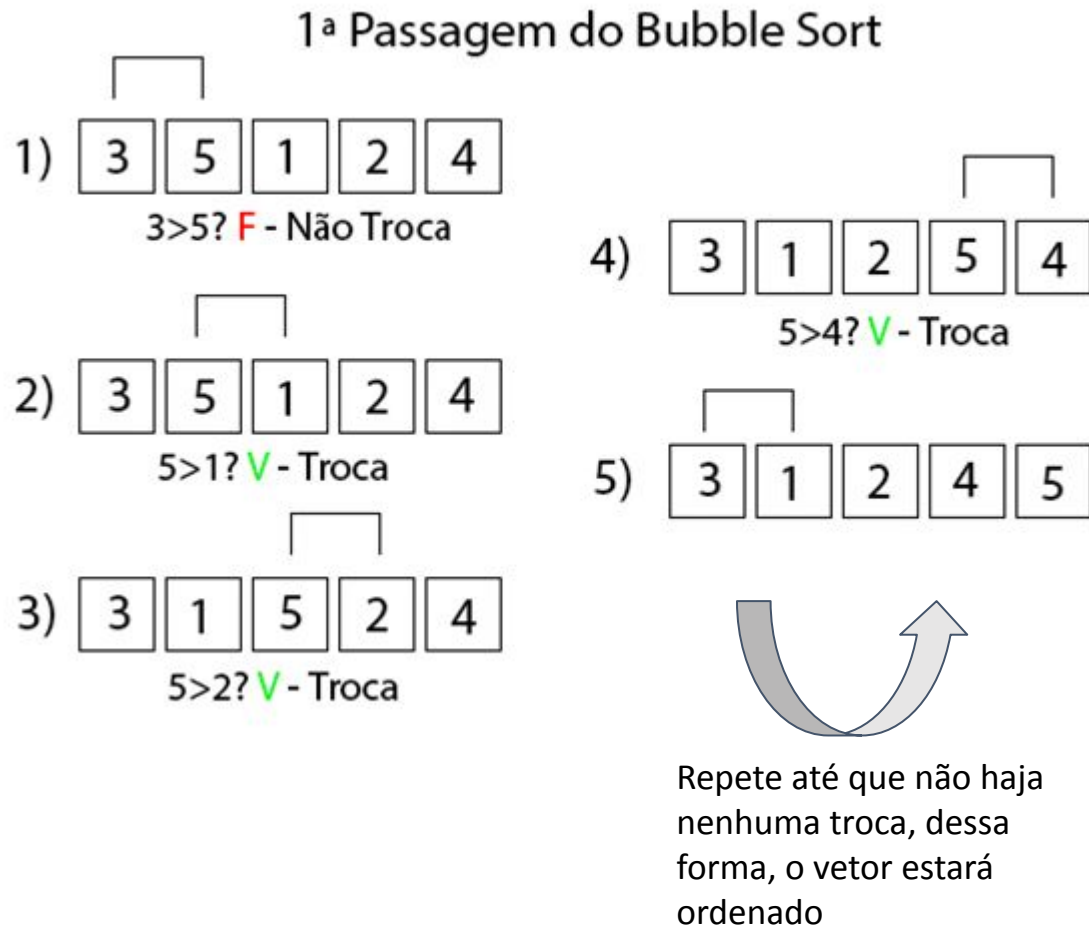


# Bubble Sort

- Algoritmo de ordenação por flutuação ou algoritmo de ordenação por bolha.
- Neste algoritmo cada elemento da posição  $i$  será comparado com o elemento da posição  $i + 1$ . Caso o elemento da posição  $i$  for maior que o da posição  $i+1$ , eles trocam de lugar e assim sucessivamente.
  - Até que mais nenhuma troca seja necessária (elementos ordenados).
- Por causa dessa forma de execução, o vetor terá que ser percorrido quantas vezes que for necessária, tornando o algoritmo ineficiente para grandes conjuntos de dados:
  - **Melhor caso:**  $O(N)$ : Dados já ordenados, passa uma vez apenas para verificar;
  - **Pior caso:**  $O(N^2)$ .

# Bubble Sort

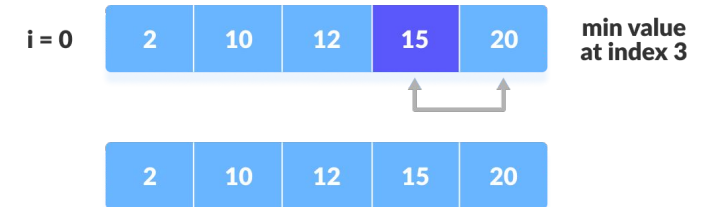
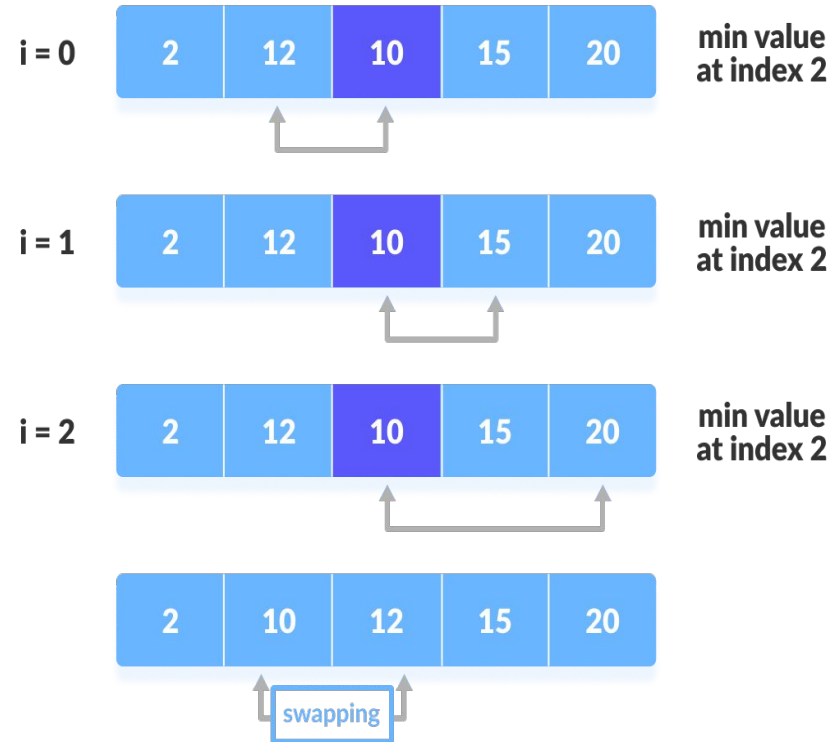
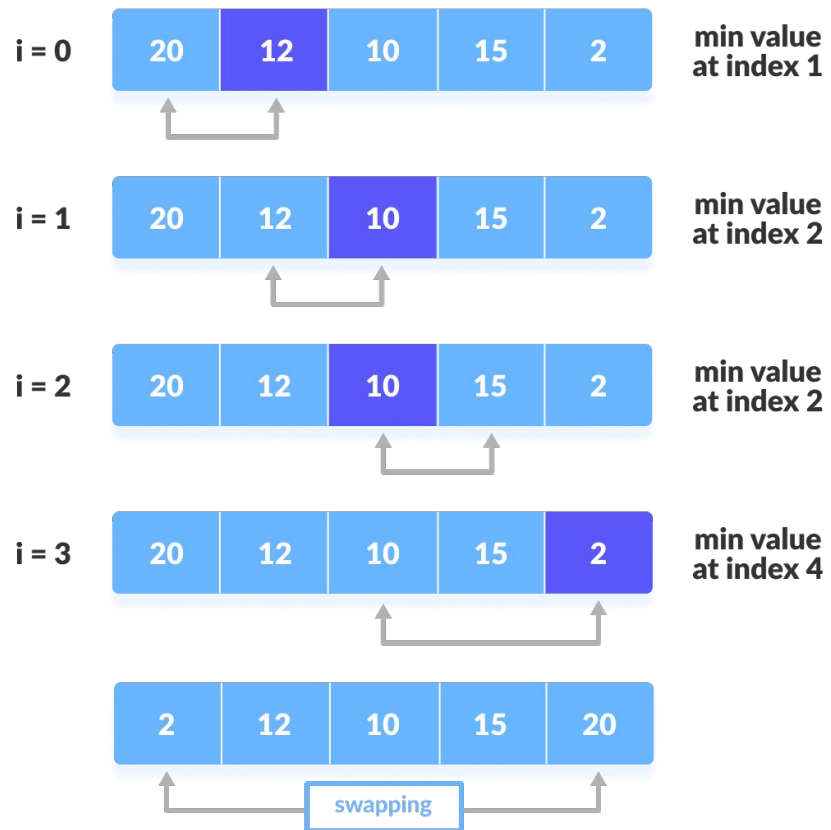
- `bool troca = False`
- 3 é maior que 5? **NÃO!**
  - **NÃO TROCA;**
- 5 é maior que 1? **SIM!**
  - **TROCA;**
  - `bool troca = True`
- 5 é maior que 2? **SIM!**
  - **TROCA;**
- 5 é maior que 4? **SIM!**
  - **TROCA;**
- Os passos acima são executados até que o vetor esteja ordenado, ou seja, nenhuma troca seja feita!



# Selection sort

- A ideia da ordenação por seleção é procurar o menor ou o maior elemento do vetor e movimentá-lo para a primeira ou a última posição do vetor.
  - Descarta-se a posição após a inserção.
  - Feito para todas as posições do conjunto de dados.
- Uma vantagem do Selection Sort é que entre os algoritmos de ordenação ele apresenta uma das menores quantidades de movimentos entre os elementos, assim pode haver algum ganho.
- Uma desvantagem é que o número de comparações é igual para o melhor caso, caso médio e o pior caso. Assim, mesmo que o vetor esteja ordenado o **custo continua quadrático ( $n^2$ )**.

# Selection sort

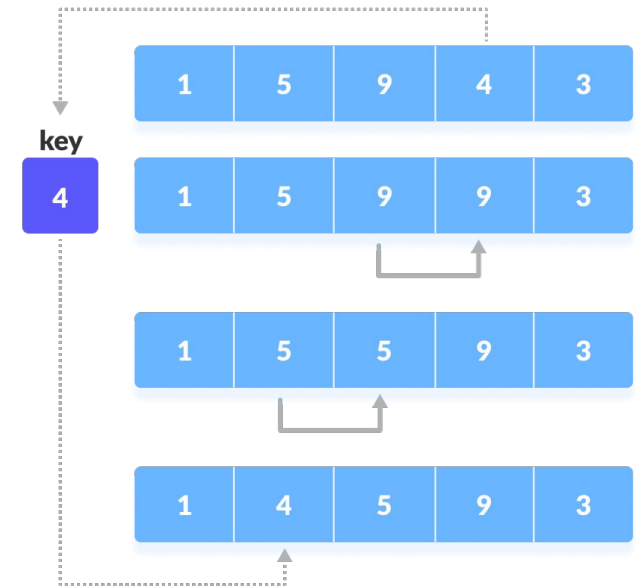
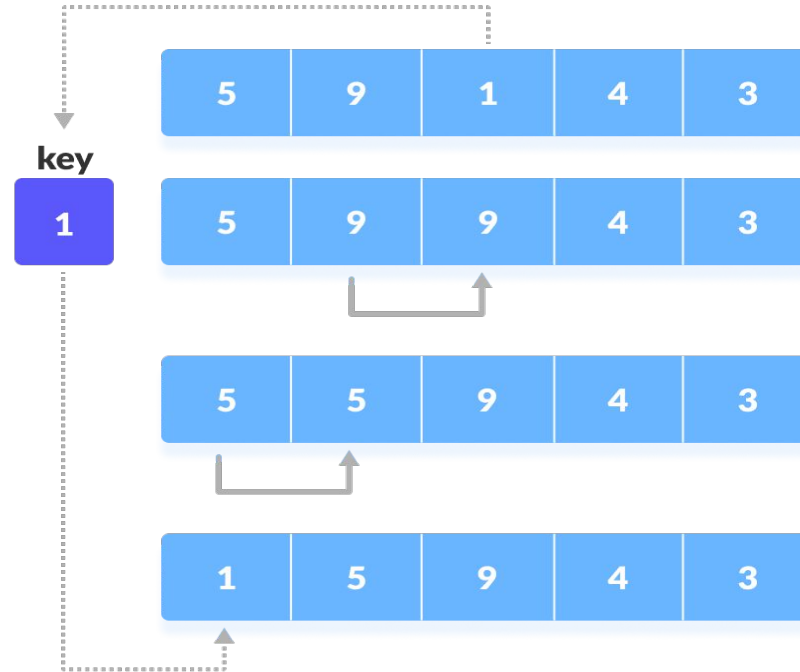
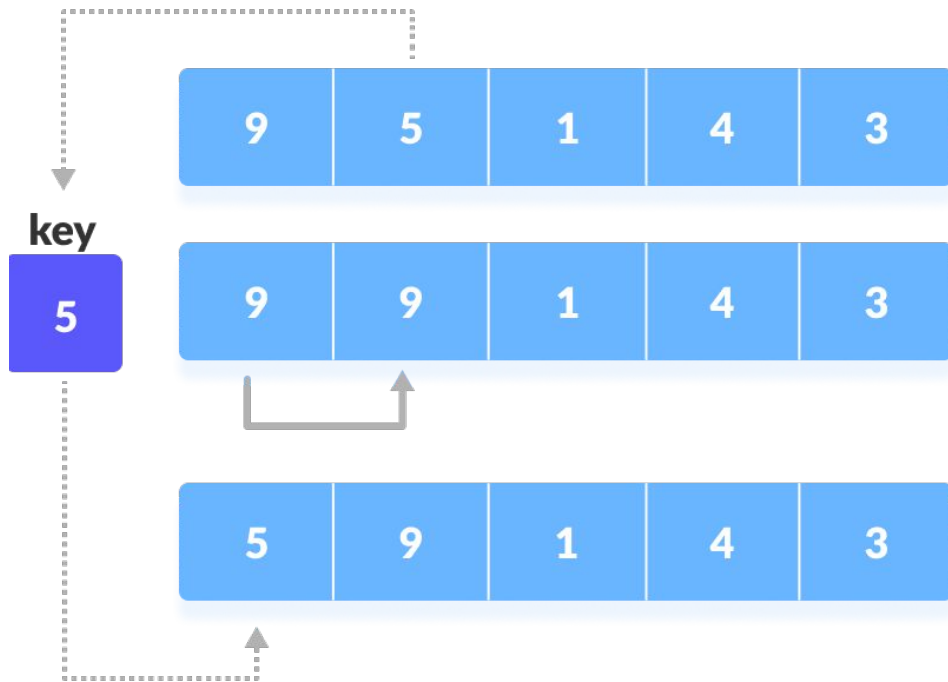


# Insertion sort

- Simples e eficiente quando aplicado em pequenas conjunto de dados.
- Neste algoritmo a lista é percorrida da esquerda para a direita, à medida que avança vai deixando os elementos mais à esquerda ordenados.
  - Estável: Não altera a ordem de dados iguais.
- A idéia da ordenação por inserção é dividir os elementos em duas subestruturas, uma com os elementos já ordenados e outra com elementos ainda por ordenar.
  - Semelhante a forma como ordenamos cartas de um baralho.
- **Melhor Caso:**  $O(N)$ .
- **Pior Caso:**  $O(N^2)$ .



# Insertion sort





# Insertion sort

