



***unipac.br***  
*Barbacena*

Bacharelado em Ciência da Computação

---

# Estruturas de Dados

## Material de Apoio

*Parte VI – Ordenação*

Prof. Nairon Neri Silva  
naironsilva@unipac.br

2º sem / 2021

# Ordenação

---

- Ordenar: processo de rearranjar um conjunto de objetos em uma ordem ascendente ou descendente.
- A ordenação visa facilitar a recuperação posterior de itens do conjunto ordenado.
  - Dificuldade de se utilizar um catálogo telefônico se os nomes das pessoas não estivessem listados em ordem alfabética.
- Atividade relevante e fundamental em processamento de dados.

# Ordenação

---

- Notação utilizada nos algoritmos:
  - Os algoritmos trabalham sobre os registros de um arquivo.
  - Cada registro possui uma chave utilizada para controlar a ordenação.
  - Podem existir outros componentes em um registro.

# Ordenação

---

- Estrutura de um registro:

```
typedef struct {  
    int Chave;  
    /* outros componentes */  
} TipoItem;
```

- Qualquer tipo de chave sobre o qual exista uma regra de ordenação bem-definida pode ser utilizado.

# Estabilidade

---

- Um método de ordenação é estável se a ordem relativa dos itens com chaves iguais não se altera durante a ordenação.
- Alguns dos métodos de ordenação mais eficientes não são estáveis.
- Ex.: Se uma lista dos funcionários é ordenada pelo campo “Salário”, um método estável produz uma lista em que os funcionários com o mesmo salário aparecem em ordem alfabética

# Estabilidade

---

- A estabilidade pode ser forçada quando o método é não-estável.
- Sedgewick (1988) sugere agregar um pequeno índice a cada chave antes de ordenar, ou então aumentar a chave de alguma outra forma.

# Estabilidade - Exemplo

*Elementos ordenados pelo código. Devendo ser ordenados pelo salário:*

código	10	20	30	40	50	60	70	80	90
salário	R\$ 100,00	R\$ 100,00	R\$ 200,00	R\$ 400,00	R\$ 500,00	R\$ 600,00	R\$ 600,00	R\$ 500,00	R\$ 400,00

*Após Ordenação...*

*Estável: ordenou pelo salário e manteve a ordem relativa dos códigos iguais.*

10	20	30	40	90	50	80	60	70
R\$ 100,00	R\$ 100,00	R\$ 200,00	R\$ 400,00	R\$ 400,00	R\$ 500,00	R\$ 500,00	R\$ 600,00	R\$ 600,00

*Instável: ordenou pelo salário, porém alterou a ordem dos códigos.*

20	10	30	90	40	50	80	70	60
R\$ 100,00	R\$ 100,00	R\$ 200,00	R\$ 400,00	R\$ 400,00	R\$ 500,00	R\$ 500,00	R\$ 600,00	R\$ 600,00

# Classificação do Métodos de Ordenação

- Classificação dos métodos de ordenação:
  - **Ordenação interna:** arquivo a ser ordenado cabe todo na memória principal.
  - **Ordenação externa:** arquivo a ser ordenado não cabe na memória principal.



# Classificação do Métodos de Ordenação

- Diferenças entre os métodos:
  - Em um método de ordenação interna, qualquer registro pode ser imediatamente acessado.
  - Em um método de ordenação externa, os registros são acessados sequencialmente ou em grandes blocos.

# Classificação do Métodos de Ordenação

---

- A maioria dos métodos de ordenação é baseada em **comparações** das chaves.
- Existem métodos de ordenação que utilizam o princípio da **distribuição**.

# Princípio da distribuição

---

- Exemplo de ordenação por distribuição:
  - considere o problema de ordenar um baralho com 52 cartas na ordem:

$A < 2 < 3 < \dots < 10 < J < Q < K$

e

$\clubsuit < \diamondsuit < \heartsuit < \spadesuit$

# Princípio da distribuição

---

- Algoritmo:
  - 1. Distribuir as cartas abertas em treze montes:  
ases, dois, três, : : :, reis.
  - 2. Colete os montes na ordem especificada.
  - 3. Distribua novamente as cartas abertas em quatro montes: paus, ouros, copas e espadas.
  - 4. Colete os montes na ordem especificada.

# Princípio da distribuição

---

- O método não utiliza comparação entre chaves.
- Uma das dificuldades de implementar este método está relacionada com o problema de lidar com cada monte.
- Se para cada monte nós reservarmos uma área, então a demanda por memória extra pode tornar-se proibitiva.

# Ordenação Interna

---

- Na escolha de um algoritmo de ordenação interna deve ser considerado o tempo gasto pela ordenação.
- Sendo  $n$  o número registros no arquivo, as medidas de complexidade relevantes são:
  - Número de comparações  $C(n)$  entre chaves.
  - Número de movimentações  $M(n)$  de itens do arquivo.

# Ordenação Interna

---

- O uso econômico da memória disponível é um requisito primordial na ordenação interna.
- Métodos de ordenação *in situ* são os preferidos.

# Ordenação Interna

---

- Métodos que utilizam listas encadeadas não são muito utilizados.
- Métodos que fazem cópias dos itens a serem ordenados possuem menor importância.



# Classificação dos métodos de ordenação interna

---

- Métodos simples:
  - Adequados para pequenos arquivos.
  - Requerem mais comparações que os métodos eficientes.
  - Produzem programas pequenos.
- Métodos eficientes:
  - Adequados para arquivos maiores.
  - Usam menos comparações.
  - As comparações são mais complexas nos detalhes.
- *Métodos simples são mais eficientes para pequenos arquivos.*

# Exercício

---

Considere que os números a seguir estão em um array do tipo int na ordem apresentada:

10 – 4 – 8 – 7 – 11 – 1 – 15 – 3 – 5 – 4 – 2 – 12 – 6

Crie um algoritmo que seja capaz de ordenar os números presentes no array sem utilizar outro array (você pode usar variáveis auxiliares).

**Obs.:** Crie a sua solução, não copie nenhum algoritmo de ordenação.

# Algoritmos de Ordenação Interna

---

- Métodos simples
  - Seleção (Selection Sort)
  - Inserção (Insertion Sort)
  - Bolha (Bubblesort)
- Métodos eficientes
  - ShellSort
  - HeapSort
  - QuickSort
  - MergeSort

# Referências

---

- FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico. *Lógica de Programação*. Makron books.
- GUIMARAES, Angelo de Moura; LAGES, Newton Alberto Castilho. *Algoritmos e estruturas de dados*. LTC Editora.
- FIDALGO, Robson. *Material para aulas*. UFRPE.
- NELSON, Fábio. *Material para aulas: Algoritmo e Programação*. UNIVASP.
- FEOFILOFF, P., *Algoritmos em linguagem C*, Editora Campus, 2008.
- ZIVIANI, N., *Projeto de algoritmos com Implementações em Pascal e C*, São Paulo: Pioneira, 2d, 2004.
- <http://www.ime.usp.br/~pf/algoritmos/>
- MELLO, Ronaldo S., *Material para aulas: Ordenação de Dados*, UFSC-CTC-INE
- MENOTTI, David, *Material para aulas: Algoritmos e Estrutura de Dados I*, DECOM-UFOP